

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки  
Кафедра автоматики та управління в технічних системах**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Дипломний проєкт**  
**на здобуття ступеня бакалавра**  
**за освітньо-професійною програмою «Комп'ютеризовані системи**  
**управління»**  
**спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»**  
**на тему: «Система управління квадрокоптером»**

Виконав:

Студент IV курсу, групи ІА-62

Клярський Кирило Андрійович \_\_\_\_\_

Керівник:

Доцент, кандидат технічних наук

Полторак Вадим Петрович \_\_\_\_\_

Рецензент:

Професор доктор технічних наук

Жураковський Богдан Юрійович \_\_\_\_\_

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_

Київ – 2020 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра автоматики та управління в технічних системах**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 151 «Автоматизація та комп'ютерно-інтегровані технології»

Освітньо-професійна програма «Комп'ютеризовані системи управління»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

**Клярському Кирилу Андрійовичу**

1. Тема проєкту «Система управління квадрокоптером», керівник проєкту Полторак Вадим Петрович, к.т.н., доцент, затверджені наказом по університету від «7» травня 2020р. №1081-с

2. Термін подання студентом проєкту 09.06.2020

3. Вихідні дані до проєкту: модель рою БПЛА, симуляції та характеристики поведінок рою; кількість агентів рою – 20, кластерів гуртування - 5, час гуртування – 15 с., максимальна швидкість при гуртуванні – 25 м/с, максимальна дальність підтримки зарядної станції – 600 м.

4. Зміст пояснювальної записки: порівняльний аналіз аналогів об'єкта дослідження і розробки. розробка моделей і структур рою дронів. розробка математичних моделей об'єкта ройового управління. розробка алгоритмів управління роєм об'єктів. моделювання управління роєм об'єктів

5. Перелік графічного матеріалу: передавальна функція, структурна схема управління, модель для слідування шляху, часові діаграми реакції системи.

6. Дата видачі завдання 7 лютого 2020 року.

Календарний план

№	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Порівняльний аналіз аналогів об'єкта дослідження і розробки	05.02.20-15.02.20	
2	Розробка моделей і структур рою дронів	15.02.20-03.03.20	
3	Розробка математичних моделей об'єкта ройового управління	03.03.20-03.04.20	
4	Розробка алгоритмів управління роєм об'єктів	03.04.20-13.05.20	
5	Моделювання управління роєм об'єктів	13.05.20-05.06.20	

Студент

Кирило КЛЯРСЬКИЙ

Керівник

Вадим ПОЛТОРАК

## АНОТАЦІЯ

Клярський К.А., Система управління квадрокоптером. КПІ ім. Ігоря Сікорського, Київ, 2020.

Проект містить 103 с. тексту, 43 рисунки, 7 таблиць та посилання на 23 літературні джерела.

Ключові слова: рій БПЛА, ройові алгоритми, симуляція, цифрова система.

Об'єктом керування є рій БПЛА.

Предметом дослідження є керування роєм БПЛА.

Метою дипломного проекту є покращення часових характеристик рою БПЛА.

У дипломному проєкті розроблено систему управління БПЛА. Розроблено математичну модель БПЛА та цифрову передавальну функцію. Розроблено кінетичну модель рою. Моделювання було реалізовано за допомогою додатка MatLab/Simulink та середовища програмування OMneT++.

Отримані результати можуть бути корисними при автоматизації аналогічних чи подібних об'єктів.

## ANNOTATION

Kliarskyi K.A. UAV control system. Igor Sikorsky Kyiv Polytechnic institute, Kyiv, 2020.

The project contains 103 pages of text, 43 figures, 7 table and links to 23 references.

Keywords: UAV swarm, multi-UAV control, swarm algorithms, simulation, digital system.

The object of control is a uav swarm.

The subject of the research is a uav swarm control system.

The purpose of this research is to increase time characteristics of uav swarm.

In the bachelor thesis is developed a UAV swarm control system. A mathematical model of UAV and digital transfer function of the control system have been developed. A kintic interaction model has been developed. The modelling has been realeased by using MatLab/Simulink and IDE OMneT++

The results obtained can be useful in automation similar objects.

Номер пальця	Форм	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
			Документація загальна			
			Знову розроблена			
	A	IA62.080BAK.002 ПЗ	Пояснювальна записка	9		
	A	IA62.080BAK.003 Д1	Математична модель	1		
	A	IA62.080BAK.004 Д2	Схема моделі управління	1		
			пакеті MATLAB/Simulink			
	A	IA62.080BAK.005 Д3	Часові діаграми реакції	1		
			системи			
		IA62.080BAK.006 Д4	Схема контролю орієнтації	1		

**Пояснювальна записка  
до дипломного проєкту  
на тему: «Система управління квадрокоптером»**

Київ – 2020 року

## ЗМІСТ

<a href="#"><u>ВСТУП</u></a>	6
<a href="#"><u>1 ПОРІВНЯЛЬНИЙ АНАЛІЗ АНАЛОГІВ ОБ'ЄКТА РОЗРОБКИ</u></a>	8
<a href="#"><u>1.1 Система керування UgCS</u></a>	8
<a href="#"><u>1.2 «Mind of the swarm»</u></a>	9
<a href="#"><u>1.3 Multi-Drone Systems</u></a>	9
<a href="#"><u>2 РОЗРОБКА МОДЕЛЕЙ І СТРУКТУР РОЮ ДРОНІВ</u></a>	11
<a href="#"><u>2.1 Загальні відомості про дрони та вибір моделі</u></a>	11
<a href="#"><u>2.2 Структура рою БПЛА</u></a>	13
<a href="#"><u>2.2.1 Походження</u></a>	13
<a href="#"><u>2.2.2 Самоорганізація</u></a>	13
<a href="#"><u>2.2.3 Позитивні та негативний зворотній зв'язок</u></a>	14
<a href="#"><u>2.2.4 Випадковість</u></a>	15
<a href="#"><u>3 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ ОБ'ЄКТА РОЙОВОГО УПРАВЛІННЯ</u></a>	16
<a href="#"><u>3.1 Ройові Алгоритми</u></a>	16
<a href="#"><u>3.1.1 Оптимізація колонії мурашок</u></a>	16
<a href="#"><u>3.1.2 Еволюційні обчислення</u></a>	17
<a href="#"><u>3.1.3 Протокол локального голосування</u></a>	18
<a href="#"><u>3.1.4 Ройовий інтелект</u></a>	20
<a href="#"><u>3.2 Розробка математичної моделі</u></a>	23
<a href="#"><u>3.2.1 Вектори швидкостей</u></a>	23

					ІА-62.080БАК.002 ПЗ			
	Арк.	№ докум.	Підпис	Дата	Система управління квадрокоптером Пояснювальна записка	Літ.	Арку	Аркуші
Розробив.	Клярськи						2	103
Перевірів	Полторак							
						КПІ ім. Ігоря Сікорського ФІОТ Гр. ІА-62		
Н. контр.								



<a href="#"><u>3.2.2 Рух у просторі</u></a>	27
<a href="#"><u>3.2.3 Опис руху матеріальної точки</u></a>	27
<a href="#"><u>3.2.4 Кутове обертання навколо нерухомої точки</u></a>	29
<a href="#"><u>3.2.5 Тензор інерції та маси квадрокоптера</u></a>	34
<a href="#"><u>3.2.6 Підсумок рівнянь для руху в 6DOF</u></a>	35
<a href="#"><u>3.2.7.1 Гравітаційні сили</u></a>	36
<a href="#"><u>3.2.7.2 Опір повітря</u></a>	36
<a href="#"><u>3.2.8 Сили від лез</u></a>	37
<a href="#"><u>3.2.9 Приводи квадрокоптера</u></a>	42
<a href="#"><u>3.2.10 Дискретні елементи</u></a>	43
<a href="#"><u>3.2.11 Математична модель квадрокоптера</u></a>	44
<a href="#"><u>4 РОЗРОБКА АЛГОРИТМІВ УПРАВЛІННЯ РОЄМ ОБ'ЄКТІВ</u></a>	49
<a href="#"><u>4.1 Алгоритм управління роєм динамічних об'єктів</u></a>	49
<a href="#"><u>4.2 Стратегії групового управління</u></a>	51
<a href="#"><u>4.2.1 Узагальнена модель інформаційної взаємодії</u></a>	52
<a href="#"><u>4.2.2 Аналіз моделі внутрішнього інформаційного взаємодії</u></a>	54
<a href="#"><u>4.3 Структура загальної кінетичної моделі рою</u></a>	55
<a href="#"><u>4.3.1 Проектування моделі рою</u></a>	56
<a href="#"><u>4.3.2 Узагальнення структура взаємодіючого рою</u></a>	58
<a href="#"><u>4.3.3 Переслідування рухомої цілі</u></a>	59
<a href="#"><u>4.3.4 Багатомісцевий пошук</u></a>	63
<a href="#"><u>4.4 Програмне забезпечення</u></a>	64
<a href="#"><u>4.4.1 X-Plane Simulator</u></a>	64
<a href="#"><u>4.4.2 OMNeT ++</u></a>	67
<a href="#"><u>4.4.3 Регулююче програмне забезпечення</u></a>	69

<a href="#"><u>4.4.4 Контролер гуртування</u></a> .....	70
<a href="#"><u>4.4.5 Контролер транспортного засобу</u></a> .....	72
<a href="#"><u>5 МОДЕЛЮВАННЯ УПРАВЛІННЯ РОЄМ ОБ'ЄКТІВ</u></a> .....	75
<a href="#"><u>5.1 Симуляція поведінки агентів</u></a> .....	75
<a href="#"><u>5.1.1 Поведінка агентів</u></a> .....	75
<a href="#"><u>5.1.2 Проектування Регулятора</u></a> .....	77
<a href="#"><u>5.1.3 Сценарії поведінки</u></a> .....	79
<a href="#"><u>5.1.3.1 Сценарій пошуку</u></a> .....	79
<a href="#"><u>5.1.3.2 Сценарій з перешкодами</u></a> .....	81
<a href="#"><u>5.1.3.3 Сценарій із прокладання заздалегідь відомого шляху</u></a> .....	82
<a href="#"><u>5.1.3.4 Планування простору симуляції</u></a> .....	85
<a href="#"><u>5.1.3.5 Сценарій кластеризації</u></a> .....	88
<a href="#"><u>5.1.3.6 Сценарій польоту в умовах обмежених ресурсів</u></a> .....	89
<a href="#"><u>ВИСНОВКИ</u></a> .....	94
<a href="#"><u>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</u></a> .....	95
<a href="#"><u>Додаток А – Код скрипта для синхронного переміщення рою</u></a> .....	97
<a href="#"><u>Додаток Б – Код Скрипту для кластеризації</u></a> .....	101

## СПИСОК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

БПЛА - Безпілотні літальні апарати

CFR CF - регулюючий орган авіації в США

FAA – Federal Aviation Administration

НСУ – Наземна Станція Управління

ПП – Процесорний Пристрій

ДІВ – Диструктивний інформаційний вплив

ІВ – Інформаційна взаємодія

YPR – yaw-pitch-roll (крен, тангаж, ривання)

ММ – Математична модель

ЦМ – Центр мас

ЕРС – Електрорушійна сила

Формування, гуртування – процес об'єднання окремих елементів рою

ToI – Тензор інерції

MoI – момент інерції

6DOF – БПЛА із 6 ступенями свободи(degree of freedom)

Зм	Лист	№ докум.	Підп.	Дата

**ІА-62.080БАК.002**

Аркуш

5

## ВСТУП

Безпілотні літальні апарати (БПЛА) мають великий потенціал для підтримки пошукових завдань в неструктурованому середовищі. Успішно знайшли вони свій шлях до цивільного застосування. За останні кілька років було розроблено та розповсюджено широкий спектр моделей, які сьогодні доступні для кінцевих споживачів. БПЛА літають маршрутами автономно, несуть камери для виконання аерофотозйомок та можуть перевозити товари з одного місця в інше. Невеликі, легкі, малошвидкісні та спритні безпілотники – такі мультимоторні дрони можуть включати багато видів датчиків, які підходять для виявлення об’єктів пошуку в місцях стихійних лих. Рій БПЛА має можливість діяти автономно під час пошуку цілей та передачі інформації усім членам рою.

Спектр застосувань широкий, включаючи повітряний моніторинг промислових підприємств, сільського господарства та підтримка у разі катастроф для швидкої оцінки ситуації та координації сил реагування на них. БПЛА можна класифікувати за їх розмірами та вагою, дальністю польоту, висотою та типом двигуна. Терміни «безпілотник» та «літаючий робот», «дрон» часто використовуються як синоніми БПЛА.

Часто вигідно, якщо використовується команда скоординованих БПЛА, а не один БПЛА. Кілька БПЛА можуть швидше охопити задану ділянку або одночасно робити фотографії з різних точок зору. Метою даної дипломної роботи було створення моделювання рою БПЛА, який має автономний контроль. Акцент був зроблен на швидкодію. Рій повинен мати реалістичний і практичний підхід для виконання необхідних місій пошуку цілі та виконання рятувальних дій. Опціонально він може бути екіпірован необхідним специфічним обладнанням.

Промисловість БПЛА орієнтується переважно на сферу послуг. Фактично самі БПЛА звичайно грають велику роль, але реальна цінність

БПЛА - це тип корисних навантажень, які вони можуть нести, і який тип послуг вони можуть ефективно надавати.

Деякі з цих випадків використання включають фотографію, кінематографію, точне землеробство, інспекцію ліній електропередач та споруд, безпеку спостереження, зйомки інфрачервоного та багатоспектрального зображення, відновлення місцевості після стихійного лиха пошук, рятувальні операції та багато іншого.

Такі системи можуть використовуватись у вітчизняній безпеці, цивільній обороні, для прикордонних застосувань, можуть допомагати Поліції, бути долученими до корпоративних операцій, допомагати при льотній підготовці та картографуванні а також сприяти пошуковим та рятунковим місіям.

Таким чином було наведено приклади використання роїв БПЛА у різних сценаріях, як і коли вони можуть бути потрібні та залучені до експлуатації. Очевидно, в усіх цих завданнях питання швидкодії стоїть доволі гостро. Іноді кожна хвилина може бути непомірно коштовною. Саме тому змодельованій системі акцент було зроблено на швидкодію.

Зм	Лист	№ докум.	Підп.	Дата

**ІА-62.080БАК.002**

Аркуш

7

# 1 ПОРІВНЯЛЬНИЙ АНАЛІЗ АНАЛОГІВ ОБ'ЄКТА РОЗРОБКИ

## 1.1 Система керування UgCS

UgCS – це комплексне рішення, розроблене «SPH Engineering», для планування маршрутів та контролем за дронами в польоті, яке може застосовуватися в багатьох дослідницьких проєктах. Автоматизоване планування маршрутів дронів з урахуванням рельєфу місцевості, вбудовані інструменти фотограмметрії з використанням геотегів, карти, політ з урахуванням заміни батарей для довгих маршрутів, роблять UgCS ефективним рішенням для професіоналів.

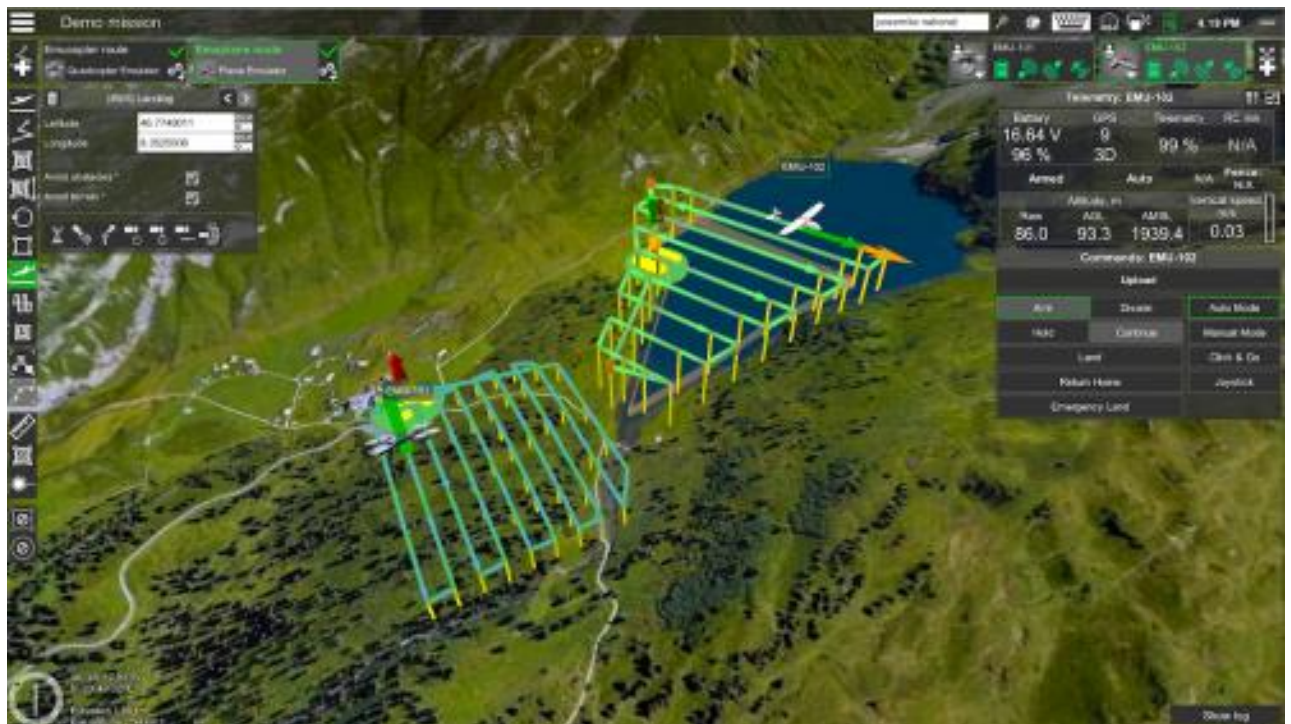


Рисунок 1.1 – Екран управління у програмі UgCS

UgCS виконує планування місій з урахуванням всіх популярних платформ БПЛА. UgCS дозволяє моніторити і контролювати один або декілька безпілотних апаратів в загальній місії, паралельно з декількох комп'ютерів.

Вартість – 39 доларів у місяць.

Недоліки: вартість.

Переваги: комплексне професійне рішення.

### 1.2 «Mind of the swarm»

Дане програмне забезпечення дозволяє БПЛА працювати разом у єдиному формуванні, яке може долати перешкоди і здатне літати в райони, занадто небезпечні для людей. Відповідно до контракту DARPA, передового дослідницького агентства Пентагону, Raytheon Technologies розробляє інструменти для автономного контролю над роями наземних і повітряних безпілотників. Використовуючи мовлення та жести, оператори можуть керувати тактикою рою серед декількох різних безпілотників через планшет чи інший пристрій.

Недоліки: вузьконаправлене рішення.

Переваги: безкоштовне.

### 1.3 Multi-Drone Systems

Команда проекту в Клагенфурті розробила платформу для систем, що передбачають співпрацю безлічі БПЛА. Одним з головних застосувань є зйомка оглядового повітряного зображення певної місцевості для підтримки служб швидкого реагування. Фотографія запезпечуються завдяки використанню високотехнологічних пристроїв та спеціальному програмному забезпеченню.

Основна операція починається з визначення інтересуючих областей на електронній карті, яка використовується для обчислення маршрутів для окремих БПЛА (планування місії).

Потім БПЛА пролітають над цією зоною та роблять зображення. Зображення надсилаються на наземну станцію (мережа) та надсилаються мозаїкою до великого оглядового зображення (сенсорний аналіз даних).

Завдяки використанню високих технологій, така система може виконувати різноманітні сценарії поведінки: такі як сценарій із перехоплення об'єкта, що рухається, здійснення високоточних маневрів, динамічне сканування місцевості і тд.

Іншим випадком використання є використання мережі БПЛА для доставки важливих товарів у віддалені райони. Виникає питання, як розподіляються робочі місця для багатьох БПЛА розподіленим способом, виходячи з попиту клієнтів та місць розташування депо.

Недоліки: приватна розробка.

Переваги: комплексне професійне рішення.

#### Висновки

1. Було приведено порівняльний аналіз вже існуючих східних рішень по забезпеченню контролю за БПЛА.
2. Як за ідею подальшої роботи було прийнято рішення взяти найкращі наработки із існуючих рішень та на їх основі відмоделювати рій БПЛА.



## 2 РОЗРОБКА МОДЕЛЕЙ І СТРУКТУР РОЮ ДРОНІВ

### 2.1 Загальні відомості про дрони та вибір моделі

Сучасний метод комерційних операцій – один пілот для управління одним БПЛА, тоді як інші члени екіпажу виступають в якості контролю за місією або візуальними спостерігачами [1]. Апаратне забезпечення, що бере участь у традиційній роботі, включає переносний передавач для керування БПЛА, пов'язаними з ними корисними навантаженнями та комп'ютером із функціонуванням програмного забезпечення наземного управління як наземна станція управління (НСУ) для напіваавтономного управління. На рисунку 2.1 показана блок-схема обладнання для традиційних комерційних застосувань.

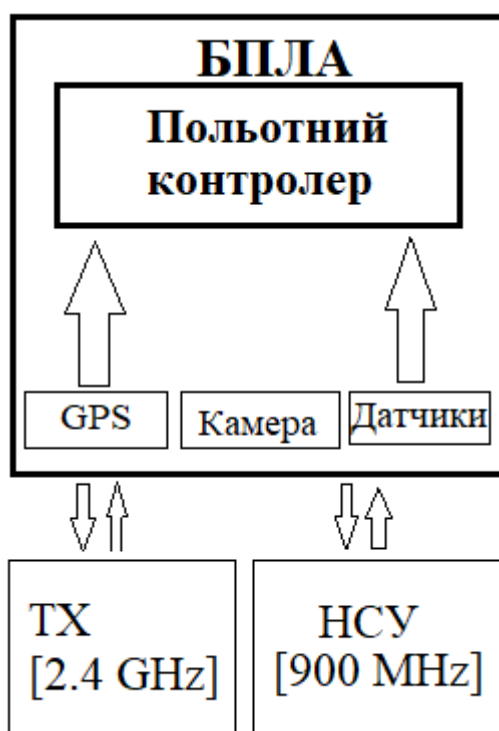


Рисунок 2.1 — Схема устаткування

Найбільш поширені квадрокоптер, гексакоптер і октокоптер:

а) квадрокоптер: ці транспортні засоби мають 4 ротора для підйому та приведення в рух і їх рух керується зміною тяги кожного ротора. Щоб

уникнути їх обертання навколо себе, два протилежних лежачих ротора крутяться в одному напрямку, а інші два навпроти лежачі ротори крутяться у зворотному напрямку.

б) кексакоптер: найбільш помітна різниця на рівні конструкції між цими представниками і попередніми - кількість гвинтів, які вони мають. Більше того, рівень їх продуктивності та стабільності значно краще, і при втраті одного двигуна, можуть здійснити посадку.

в) октокоптер: вони приводяться в рух вісьмома роторами, і вони працюють аналогічно, як і вони попередні. Власне, політ може продовжитися і після втрати одного двигуна або два з протилежних сторін. В основному вони використовуються для перевезення важких предметів через їх велику стійкість.

Таблиця 2.1 – Порівняння різних типів дронів в залежності від кількості моторів

Характеристика	Квадракоптер(4)	Гексакоптер(6)	Октокоптер(8)
ЦІНА	НИЗЬКА	СЕРЕДНА	ВИСОКА
ВАРТІСТЬ РЕМОНТУ	НИЗЬКА	СЕРЕДНА	ВИСОКА
МАНЕВРІНІСТЬ	ВИСОКА	СЕРЕДНА	НИЗЬКА
ДОДАТКОВЕ НАВАНТАЖЕННЯ	НИЗЬКЕ	СЕРЕДНЄ	ВЕЛИКЕ
РОЗМІРИ	МАЛИЙ	СЕРЕДНІЙ	ВЕЛИКИЙ

Виходячи з даної таблиці та ураховуючи специфіку винонуваної роботи були прийнято рішення обрати в якості екземпляру об'єкта дослідження дрони із чотирма двигунами.

## 2.2 Структура рою БПЛА

Опис рою починається із наведення характерних рис таких систем. Приклади таких систем можна навести із дикої природи, де ройові принципи дуже поширені.

### 2.2.1 Походження

Герардо Бені, професор електротехніки в Університеті Каліфорнії в Ріверсайді, почав використовувати термін "рій" для опису своєї роботи зі стільниковими роботами. Бені винайшов що "рій" точно описує стільникові автомати. Група клітин у роботів демонструє подібні характеристики, виявлені в таких біологічних роях, як і комахи, – такі як децентралізація та відсутність синхронізації. Рой правильно описує робототехнічні системи, але поки робототехніки зосереджуються на виконанні завдань із роєм, біологи розглядають концепцію рою та аналізують соціальну поведінку комах та інших тварин [1].

### 2.2.2 Самоорганізація

Біологи досліджують ідею, що рій може виявляти закономірність у своїй системі та самоорганізується, щоб знайти оптимальні засоби досягнення мети. Роботики намагаються створити моделі поведінки, щоб клітини самоорганізувалися. Зв'язок є ключовим параметром, який дозволяє членам взаємодіяти та самоорганізовувати свої завдання. Такі характеристики рою поширюються від децентралізації, відсутності синхронізації та простоти спілкування між членами. У справжніх роях, таких як бджоли та мурахи, члени групи можуть використовувати пряме

спілкування між собою або непряме спілкування, яке називається стигмерією, для взаємодії через навколишнє середовище [2].

### 2.2.3 Позитивні та негативний зворотній зв'язок

Бджоли та мурахи постійно шукають джерел їжі для своїх колоній для підтримки існування, тому розташування джерела їжі є загальним прикладом соціальної поведінки і легким для демонстрування за допомогою експериментів. Їх здатність встановлювати шляхи до джерела харчування ілюструє один рівень самоорганізації. Бджоли безпосередньо повідомляють місцезнаходження джерела їжі через танці. Бджолиний танець покаже відстань та напрямок джерела їжі до інших. Коли одне джерело їжі перевершує якість, бджоли віддають перевагу кращому джерелу по кількості бджіл, які танцюють, щоб вказати на це місце. Експерименти показують, що відмінності в частоті танців і відмові створюють позитивний зворотній зв'язок і викликають більше бджіл до найкращого по якості шляху. Не всі бджоли збиратимуться до найвищого джерела їжі [3]. Невелика кількість все ж таки буде продовжувати йти до альтернативного джерела їжі або до іншого місця. Таку поведінку можна розцінювати як реакцію на негативні відгуки в системі. Негативний зворотний зв'язок генерується через насичення, виснаження, скупчення та конкуренцію на джерелі їжі.

Мурахи найбільш відомі тим, що спілкуються через навколишнє середовище з феромонами для позначення якості шляху [4]. Ці феромони запроваджують метод позитивного зворотного зв'язку для колонії для посилення сліду. Постійне посилення сліду переконує інших мурашок продовжувати йти тим самим шляхом, забезпечуючи позитивні зв'язки. Негативний зворотний зв'язок вводиться через ті ж загальні ускладнення, що і бджоли: насичення, виснаження, скупченість та конкуренція на джерелі їжі.

#### 2.2.4 Випадковість

І бджоли, і мурахи покладаються на випадковість у поведінці особин у групі. Це може здатися протиінтуїтивним вважати, що самоорганізація створюється через випадковості, але випадковість дозволяє вводити в групу нові ідеї. Вона може просто запропонувати нові шляхи до джерела їжі або більш загальні нові методи та рішення, що сприяють зростання колонії. Випадковість також є джерелом оптимізації. Наприклад, два джерела їжі, які однакові за якістю і на однаковій відстані від бджолиного вулика, повинні бути використані симетрично. Експериментально відхилення кількох бджіл спричинить замах на одне джерело, тому що ті кілька бджіл набирають більше бджіл, а ті продовжують набирати ще більше. Цей же принцип стосується мурашок. Оскільки більше мурашок йде по шляху феромону, сила стає більшою. Цей шлях посилюється і стає оптимальним [3, 4].

Кілька взаємодій відбуваються по всій групі, викликаючи більше дій та реакцій. Характеристики системи позитивного зворотного зв'язку, негативного зворотного зв'язку та випадковості забезпечують баланс, необхідний для того, щоб група відповідала на постійно мінливі зміни навколишнього середовища.

#### Висновки

1. В розділі було приведено узагальнену модель комунікації.
2. Через розглядання загальних характеристик БПЛА було обрано дрона із 4 моторами як найбільш оптимальний.
3. В розділі було виведено основні принципи чому рій у природі є настільки ефективними. Подальша розробка вестиметеса орієнтовуючись на ці принципи.

## 3 РОЗРОБКА МАТЕМАТИЧНИХ МОДЕЛЕЙ ОБ'ЄКТА РОЙОВОГО УПРАВЛІННЯ

### 3.1 Ройові Алгоритми

Ройові алгоритми вважаються ефективним рішенням для розв'язання задач, що потребують залучення великої кількості агентів та загального управління [5]. Поштовх для розуміння функціонування таких алгоритмів можна отримати із природи на прикладі деяких тварин.

#### 3.1.1 Оптимізація колонії мурашок

Поведінка мурах у колонії надихала експерименти та зрештою втілила у життя алгоритми, що імітують мурах. Основним завданням колонії мурашок є пошук джерел їжі. Виконуючи це завдання, мурахи здатні знайти найкоротший шлях до цього джерела. Ця природна оптимізація випробувана Денебургом за допомогою експериментів з мостом.

Експеримент з бінарним мостом складається з двох шляхів однакової довжини від їх гнізда до місця джерела їжі, тому спочатку всі мурахи вибирають випадковий шлях. Коли вони продовжують обирати випадково, мурахи рухаються однією стежкою, та інтенсивність феромону на цьому шляху зростає. Коли незначна кількість мурашок порушує шанси 50/50 на пошук вірного вляху, вони додають більше мурашок, поки більшість мурах не буде зосереджена на одному шляху.

Таким чином можна пояснити випадковість у їх поведінці. Дві доріжки однакової довжини можна розбити на довший і короткий шлях. Денебург також створив міст від гнізда до їжі з двома довгими гілками, показаними на рисунку 3.1.

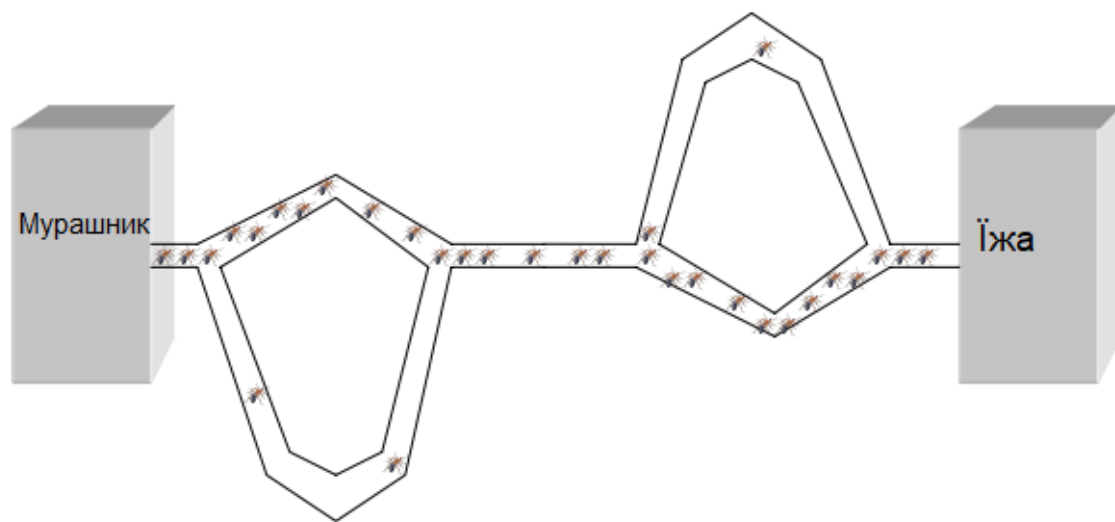


Рисунок 3.1 – Експеримент із мостом

Знову ж таки, всі шляхи спочатку вибираються випадковим чином. Мурахи, які беруть найкоротше шлях до джерела їжі та назад до гнізда очевидно спочатку повертаються до гнізда. Маршрут їх вдвічі посилений феромонами, ніж ті, що інші, тому що інші мурахи на довгих стежках пройшли через свій маршрут лише один раз. Ця початкова різниця посилюється, поки більшість мурах не піде по найкоротшому шляху [5].

### 3.1.2 Еволюційні обчислення

Генетичний та еволюційний алгоритми - це два приклади алгоритмів, які використовують набір популяції щоб виявити рішення. Ці алгоритми схожі за концепцією, але виконуються по-різному [5]. Відмінності стають нерозбірливими у більшості випадків, і генетичні і еволюційні алгоритми майже взаємозамінні за значенням. Обидва імітують природну еволюцію, використовуючи виживання найбільш придатних особин та дозволяючи маніпулювати населенням. Кожній людині в популяції присвоюється відповідна цінність залежно від проблеми. Збираючи індивіда з більш високою придатністю, населення просунеться до рішення заданої проблеми.

Різниця полягає в процесі відтворення, переміщення та мутації. Вони показують, наскільки "близько" особина на шляху до цілі, таким чином, популяція «шукає» простір для вирішення проблеми.

### 3.1.3 Протокол локального голосування

Протокол локального голосування (алгоритм типу стохастичною апроксимації) був запропонований для завдання досягнення консенсусу в мережі агентів [6]. В алгоритмі стохастичною апроксимації з постійним розміром кроку досліджувався в завданні з нелінійною динамікою в умовах мінливої топології, перешкод і затримок.

Протокол локального голосування був модернізований для задач з вартісними обмеженнями на передачу інформації в каналах зв'язку між агентами. Це завдання є окремим випадком оптимізації нестационарного функціоналу типу середнього ризику. Для управління кожен агент  $d_i \in D$  в момент часу  $t = 0, 1, 2, \dots$  має зашумлену інформацію про свій власний напрямку руху, посилену (домножену) на поточний потенціал свого маршруту:

$$y_t^{i,i} = g_t^i + \omega_t^{i,i}, \quad (3.1)$$

$$g_t^i = q_t^i x_t^i, \quad (3.2)$$

і, якщо  $D_t^i \neq \emptyset$ , зашумлені спостереження про напрямки руху сусідів, також домножені на потенціали маршрутів сусідів:

$$y_t^{i,j} = g_{t-h_t}^j + \omega_t^{i,j}, \quad j \in D_t^i, \quad (3.3)$$

де  $\omega_t^{i,j}, \omega_t^{i,i}$  – шум, а  $0 \leq h_t^{i,j} \leq \bar{h}$  – цілочисленна затримка,  $\bar{h}$  – максимально можлива затримка.



Припустимо  $\omega_t^{i,j} = 0$  та  $h_t^{i,j} = 0$  для решти всіх пар  $(i,j)$ , для яких вони не були визначені. Так як система починає роботу при  $t = 0$ , то неявна вимога до безлічі сусідів:  $j \in D_t^i \rightarrow t - h_t^{i,j} > 0$ .

Консенсусне мультиагентне управління, що формується за «протоколу локального голосування», задається наступним співвідношенням [7]:

$$u_t^{i,j} = \alpha \sum_{j \in D_t^i} b_t^{i,j} (y_t^{i,j} - y_t^{i,i}), \quad (3.4)$$

де  $\alpha$  - розмір кроку протоколу управління,  $\overline{D}_t^i \subset D_t^i, b_t^{i,i} > 0 \quad \forall \overline{D}_t^i$ . Припустимо,  $b_t^{i,i} = 0$  для інших пар  $(i,j)$ . Динаміка зміни напрямку руху об'єкта буде описуватися різницеvim рівнянням:

$$x_{t+1}^i = x_t^i + f(u_t^i, x_t^i), \quad (3.5)$$

з управлінням  $u_t^i \in \mathbb{R}$ , вплив якого на зміну напрямку  $x_t^i$  визначається деякою функцією  $f(\cdot, -): \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$ , яка формує остаточне управління відповідно до процедури уникнення зіткнень.

Як приклад застосування алгоритму розглянемо наступну задачу: переміщення рою агентів з точки А в точку В в умовах невизначеності щодо перешкод, що зустрічаються на шляху прямування [8](рисунок 3.2).



Рисунок 3.2 – Переміщення рою з точки А в точку В по протоколу локального голосування

Вплив потенціалу на рух рою полягає в тому, що при його істотній зміні, об'єктам рою необхідно змінити свій напрямок в напрямі, який підвищує ймовірність виконання завдання. Зокрема, коли рій зустрічає перешкоду, у найближчого до неї об'єкта значно падає потенціал руху по обраному ним шляху. В результаті, рою необхідно змінити свій напрямок руху, що може бути досягнуто шляхом застосування протоколу локального голосування, розглянутого вище.

### 3.1.4 Ройовий інтелект

Самоорганізація групи на впорядковані зразки є інтелектуальною характеристикою. Щоб рій формував впорядковані візерунки, під час пошуку потрібно «аналізувати» шаблони оптимальних методів. Ця характеристика могла б дозволити рою мати «інтелект». Вчений Герадо Бені боровся з визначенням роевого інтелекту, оскільки слово інтелект також настільки вільно використовується. Одне визначення розумного рою, за словами Бені, це «група машин, здатних до непередбачуваних розрахунків». Згодом він перетлумачив визначення машин як «сутності, здатної до механічної поведінки, тобто передача та / або переробка речовини / енергії » [9].

Сотні і навіть тисячі неінтелектуальних машин можуть складати інтелектуальний рій. Є переваги наявності простих компонентів у групі порівняно з складними централізованими компонентами. Шляхом самоорганізації та ідентифікації моделей, окремі машини, які працюють разом як рій, можуть виконувати завдання, які не могли бути можливі однією машиною.

Звісно, окремі члени рою будуть легше проектуватись та будуватись, тому потенційно ці прості компоненти можуть бути дешево замінені, або утилізовані. Непередбачувана функція рою походить від методу, якому він

«вчиться». Існують численні алгоритми, які дозволяють рою обчислити можливий процес для виконання різних завдань.

Так само координується рій БПЛА одиницею БПЛА, яка виконує бажане завдання або набір завдань. Суть ройового інтелекту полягає в тому, що рій, що складається з динамічних об'єктів (агентів), володіє великими можливостями в порівнянні з його окремими індивідами [10].

За допомогою встановлення комунікацій між членами рою і взаємодії із зовнішнім середовищем можливе вирішення завдань, на виконання яких у окремого агента не вистачає можливостей.

Розглянемо набір критеріїв, які є відмінними рисами об'єктів ройової системи:

- агенти (роботи) є автономними, тобто здатні рухатися і взаємодіяти з навколишнім середовищем без централізованого управління;
- задача повинна виконуватися в сукупності великою кількістю об'єктів, це означає, що система повинна бути розроблена з урахуванням масштабованості;
- рій складається з однорідних груп агентів (роботів), упор робиться на велику кількість однакових об'єктів, ніж на централізовано-керовані гетерогенні об'єкти, де кожному агенту (роботу) призначена «персональна» роль.

В цьому випадку можна розглянути три можливі стратегії управління:

- централізована - дистанційне керування з виділеної базовою станцією, лідер рою призначається з центрального вузла;
- децентралізована - лідер рою визначається на основі будь-якого алгоритму і не залежить від центральної керуючої станції [11];

– змішана - поєднує в собі переваги централізованої і децентралізованої стратегій шляхом виділення лідера рою на основі одного з алгоритмів з передачею прав управління оператору при необхідності [10, 11].

Досить численну кількість  $D$  динамічних об'єктів (агентів)  $d_i$  ( $i = 1, \dots, n$ ), спільне взаємодія яких забезпечує вирішення деякого безлічі завдань  $P = p_1, \dots, p_m$ , будемо називати роєм. При цьому вводяться наступні припущення:

- всі агенти  $d_i$  ( $i = 1, \dots, n$ ) однакові.
- агент  $d_i \in D$  може здійснювати обмін повідомленнями з деяким підмножиною агентів  $D_i \in D$ , які перебувають в межах певної зони, обмеженої радіусом  $R$ , яку зазвичай називають «зоною видимості».

За допомогою такого інформаційного обміну агенту  $d_i$  може бути доступна інформація про стан агентів підмножини  $D_i$ .

- Агент  $d_i$  рухається на відстані  $r$  від своїх найближчих сусідів.
- Для вирішення поставлених завдань  $p_i \in P$  оператор надає кожному агенту  $d_i$  карту потенціалів, визначальну перспективність руху в певному напрямку.

Однак, у агентів немає відомостей про наявність перешкод на шляху прямування. У міру руху у кожного агента формується власна потенційна картина світу.

- рух кожного агента  $d_i$  характеризується напрямком  $x_i$  і значенням потенціалу обраного шляху  $q_i$ .
- кожен агент має базу знань про навколишній світ, яку він поповнює по шляху проходження (на основі датчиків, методів комп'ютерного зору, інформації від сусідів, перерахунку значення потенціалу шляху і т. д.).
- потенційна функція характеризує можливість досягнення мети  $s$  при збереженні руху в напрямку  $x_t^i$ . Наприклад, при відомому напрямку до цілі  $s_t^i$  для агента  $i$  в момент часу  $t$  можна покласти [12]:

$$q_t^i = q_t^i(x_t^i) = \langle x_t^i, s_t^i \rangle, \quad (3.6)$$

— кожен агент при остаточному визначенні напрямку руху намагається вибрати маршрут так, щоб уникнути зіткнень.

Наприклад, можна задати функцію  $\phi(x_t^i)$ , яка відхиляє рух у випадковому напрямку в тому випадку, якщо агент  $i$  виявляє перешкоду в напрямку  $x_{it}$  на відстані ближче  $r$ .

### 3.2 Розробка математичної моделі

Загальна розробка схеми керування починається із математичної моделі, через яку можна буде сформулювати передатну функцію. Така модель конструюється на основі опису сукупних сил, діячих на об'єкт.

#### 3.2.1 Вектори швидкостей

Квадрокоптер має 6 ступенів свободи, це означає, що він має лінійні та кутові рухи. Цей складний рух (званий вільним рухом) може бути повністю визначений двома векторами: вектором положення  $\overrightarrow{pst}$  та вектором орієнтації  $\overrightarrow{ort}$ . Вектор положення має поточне положення квадрокоптера в системі координат Землі, а вектор орієнтації поточні кути орієнтації квадрокоптера порівняно з системою координат Землі.

Для розрахунку поточних значень  $\overrightarrow{pst}$  слід знати такі параметри: вектор лінійної швидкості  $\vec{v}$ , вектор лінійного прискорення  $\vec{a}$ . Для розрахунку поточних значень  $\overrightarrow{ort}$  слід знати такі параметри: вектор кутової швидкості  $\vec{\omega}$ , вектор кутового прискорення  $\vec{\epsilon}$ . Також початкові умови всіх 6 вищезгаданих векторів повинні бути відомі.

Крім того, зовнішні сили і крутні моменти, які можна було б замінити силою рівнодіючою сил  $net\vec{F}$  і крутним моментом  $net\vec{\tau}$ , призводять до зміни лінійного та кутового прискорень квадрокоптера і ці зміни впливають на

положення та орієнтацію. Отже, для створення математичної моделі (ММ) рівнянь, для векторів обчислення  $\vec{pst}$  та  $\vec{ort}$ , слід визначити:(рисунок 3.3)

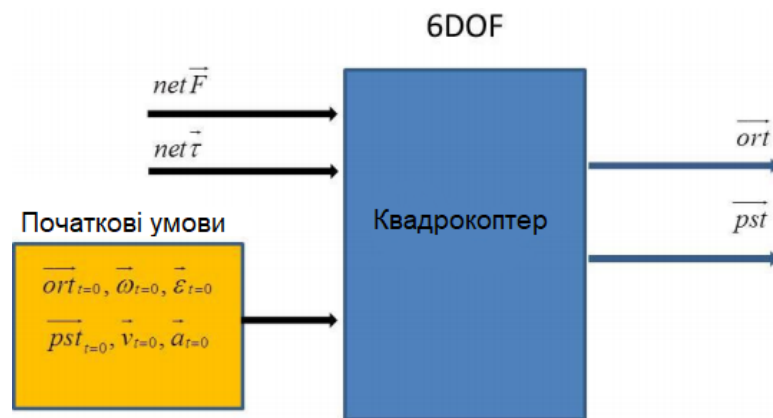


Рисунок 3.3 – Логічна схема для обчислення положення та орієнтації квадрокоптера

Існує три джерела зовнішніх сил, такі як гравітаційне поле, аеродинамічний опір і обертання лопатей у повітрі. Вони породжують гравітаційну силу  $\vec{F}_{mg}$ , силу тяги  $\vec{F}_{drag}$ , сила тертя  $\vec{T}$  відповідно. Зовнішній крутний момент  $\vec{H}$  створюється тільки шляхом обертання лопатей(рисунок 3.4).

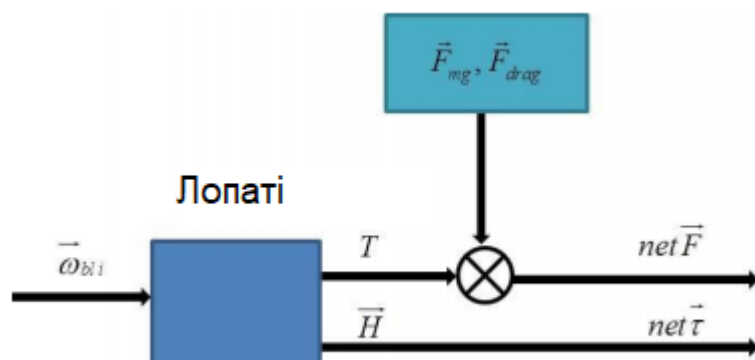


Рисунок 3.4 – Логічна схема для розрахунку рівнодіючої сил та рівнодіючої крутного моменту

Усього є чотири лопаті та чотири двигуни BLDC. Припустимо, що мотори є пронумеровано від 1 до 4. Кожен лопать обертається відповідним двигуном з частковою кутовою швидкістю  $\overrightarrow{\omega_{bli}}$ , де індекс  $i$  вказує номер мотора. Кутова швидкість валу двигуна регулюється силовим мостом і кожен силовий міст регулюється мікроконтролером. (рисунок 3.5)

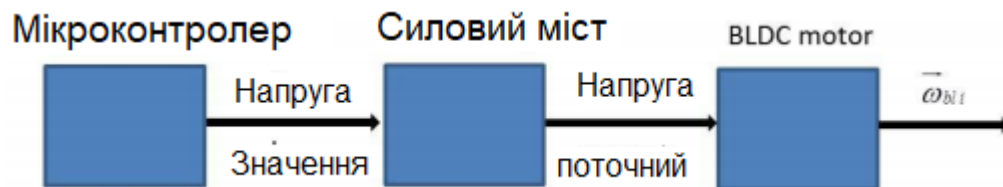


Рисунок 3.5 – Логічна схема для розрахунку кутових швидкостей лопатей

Таким чином весь процес переміщення квадрокоптера в 6DOF можна описати так: мікроконтролер задає сигнали (аналогічні або цифрові), які передаються через силові мости та двигуни до кутових швидкостей лопатей. Поворот лопаті створюють сили і крутні моменти, які разом з гравітаційними і тягловими силами змінюють положення та орієнтацію квадрокоптера (рисунок 3.6).

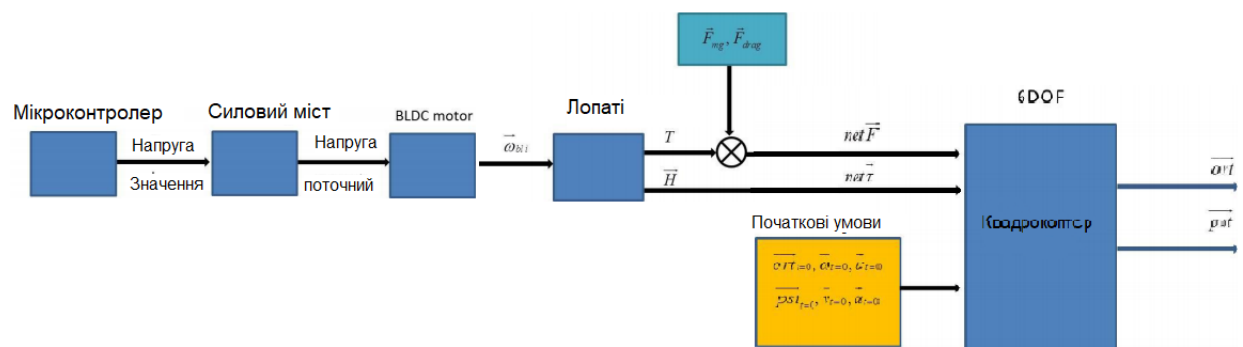


Рисунок 3.6 – Логічна схема для створення ММ квадрокоптера

Контролеру потрібна інформація зворотного зв'язку, така як положення, орієнтація та відповідні параметри (наприклад, лінійна кутова

швидкість), яку слід вимірювати датчиками. Виходячи з бажаних значень положення, орієнтації та значення зворотного зв'язку, виміряного датчиками, розроблені контролером, повинні генерувати відповідні значення для мостів потужності (рисунок 3.7).

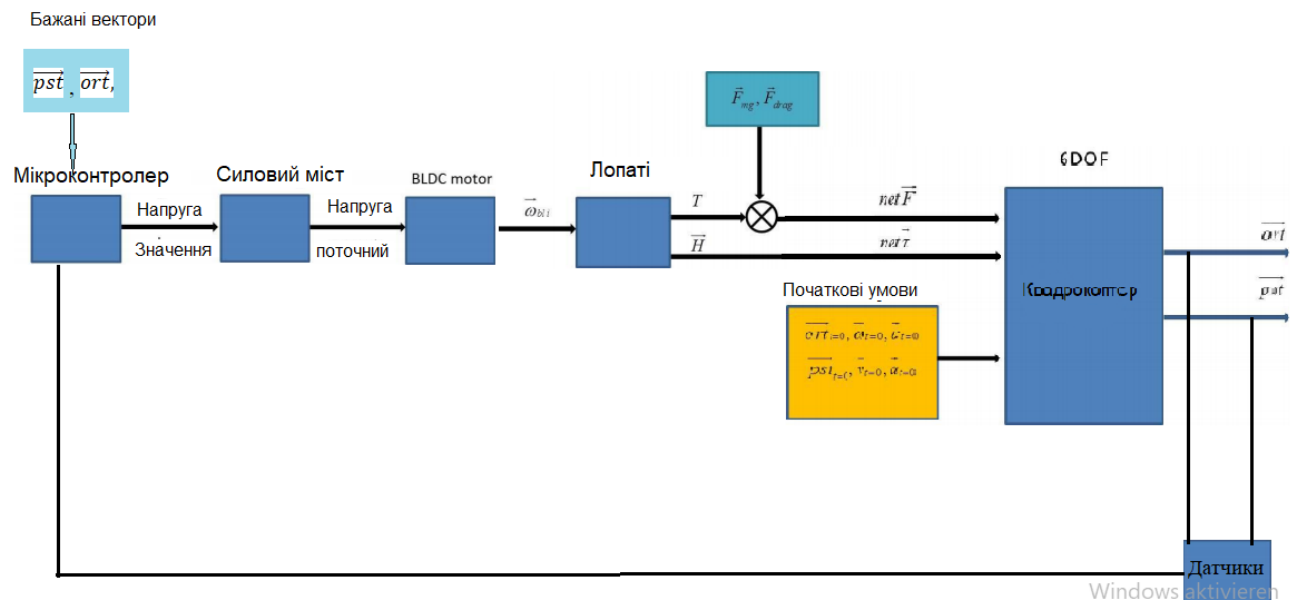


Рисунок 3.7 – Логічна схема для ММ та контролера

Можна зробити висновок, що ММ складається з передавальних функцій, які описують або оцінюють елементи та процеси, показані на рисунку. 3.8. Отже, поведінка квадрокоптер 6DOF повинна бути описана диференціальними рівняннями. Тоді слід зазначити співвідношення між кутовими швидкостями лопатей, рівнодіючою сил і обертовим моментом.

Зовнішні сили, якими неможливо керувати, гравітаційне та сила тертя повинні, бути обговорені. Елементи, необхідні для створення сили і крутного моменту для переміщення: аналогічні елементи (лопаті, двигуни, мости живлення) та цифрові елементи (датчики, мікроконтролер) повинні бути описані.

ММ повинен усереднити процес, показаний на Рисунку 3.7. Опис цього процесу включає вільний рух квадрокоптера, вплив від застосованій сил, як



відбувається обертання лопатей та наслідки дій цифрових елементів датчиків, мікроконтролер).

### 3.2.2 Рух у просторі

Для опису вільного руху квадрокоптера у просторі (процес показаний на рис. 3.7), використовується теорія вільного руху жорсткого тіла. Відповідно до цього, вільний рух твердого тіла розглядається як складний рух, який складається з двох простих рухів:

- передача руху точки з масою дорівнює масі тіла (точкова маса), де точка - будь-яка точка тіла;
- кутове обертання тіла навколо нерухомої точки, де точкова маса обрана вище, вважається фіксованою.

Передача руху точки описуються наступним чином:

$$\text{net}\vec{F} = \frac{d\vec{p}}{dt}, \quad (3.7)$$

де  $\text{net}\vec{F}$  – рівнодіюча сил всіх зовнішніх сил, які прикладені до точкової маси,  $\vec{p}$  – лінійний момент точкової маси, і  $\frac{d}{dt}$  – диференціальний оператор. Кутове обертання тіла навколо нерухомої точки можна описати як

$$\text{net}\vec{\tau} = \frac{d\vec{L}}{dt}, \quad (3.8)$$

де  $\text{net}\vec{\tau}$  – векторна сума всіх зовнішніх крутних моментів,  $\vec{L}$  - це кутовий імпульс тіла.

### 3.2.3 Опис руху матеріальної точки

Для опису руху передачі точкової маси повинна бути обрана фіксована система відліку. У деякій довільній точці простору, зазначеній як "О", обрана

фіксована система відліку XYZ. Положення точкової маси в системі відліку XYZ може бути описується радіус-вектором  $\vec{r}$  (рисунок 3.8).

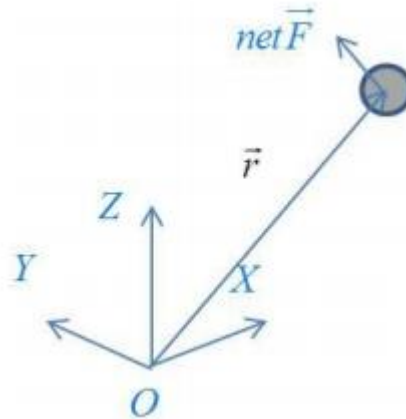


Рисунок 3.8 – Положення точки точки у фіксованій системі відліку XYZ

Щоб застосувати рівняння (3.7) до цієї точкової маси, повинен бути описан її лінійний імпульс:

$$\vec{p} = m \cdot \vec{v}, \quad (3.9)$$

Де  $m$  - маса точки і  $\vec{v}$  - лінійна швидкість точкової маси. З урахуванням рівн. (3.9), рівняння. (3.7) можна переписати як:

$$\vec{netF} = m \cdot \frac{d\vec{v}}{dt} = m \cdot \frac{d^2\vec{r}}{dt^2} \quad (3.10)$$

Зміна положення точки може бути виражена з рівняння. (3.10) як:

$$\vec{\ddot{r}} = \frac{\vec{netF}}{m}, \quad (3.11)$$

Скалярна форма рівняння (3.11):

$$\begin{aligned} \ddot{r}_X &= \frac{\vec{netF}_X}{m} \\ \ddot{r}_Y &= \frac{\vec{netF}_Y}{m}, \\ \ddot{r}_Z &= \frac{\vec{netF}_Z}{m} \end{aligned} \quad (3.12).$$

### 3.2.4 Кутове обертання навколо нерухомої точки

Як було сказано вище, для опису вільного руху тіла, кутового слід враховувати обертання навколо нерухомої точки. Припустимо фіксовану систему відліку XYZ з початком у нерухомій точці жорсткого тіла (рисунок 3.9).



Рисунок 3.9 – Обертання жорсткого тіла навколо нерухомої точки  
внерухомій системі відліку XYZ

Обертання жорсткого тіла з випадковою формою навколо нерухомої точки може бути описані у фіксованій системі відліку XYZ . Для використання рівн. (3.8) слід обчислити кутовий імпульс тіла.

Тіло розглядається як система точкових мас, де маса кожної точки  $dm$  і положення будь-якої з цих точок можна визначити вектором  $\vec{\rho}$  починаючи від початку координат XYZ до елементу  $dm$ . У цьому випадку імпульс кута дорівнює:

$$\vec{L} = \iiint \vec{\rho} \times \overrightarrow{v_{dm}} * dm \quad (3.13), [13]$$

Де  $\iiint$  - інтеграл об'ємного тіла,  $\overrightarrow{v_{dm}}$  - лінійна швидкість конкретного елемента. Рівняння (3.13) у не векторній формі виглядатиме як:

$$\begin{aligned}
L_x &= \iiint (y * v_z - z * v_y) * dm \\
L_y &= \iiint (z * v_x - x * v_z) * dm, \\
L_z &= \iiint (x * v_y - y * v_x) * dm
\end{aligned}
\tag{3.14}$$

де  $x, y, z$  - координати елемента  $dm$  (або іншими словами координати відповідного радіус вектора  $\vec{\rho}$ ),  $v_x, v_y, v_z$  - проекції швидкості цього елемента.

Лінійна швидкість кожного елемента  $\vec{v}_{dm}$  можна описати як:

$$\vec{v}_{el} = \vec{\omega} \times \vec{\rho} = \begin{pmatrix} \omega_y * z - \omega_z * y \\ \omega_z * x - \omega_x * z \\ \omega_x * y - \omega_y * x \end{pmatrix}, \tag{3.15}, [13]$$

де  $\vec{\omega}$  - кутова швидкість тіла  $i$ ,  $\omega_x, \omega_y, \omega_z$  - проекції на вісі XYZ.

Переписавши рівняння. (3.13) з урахуванням (3.15) імпульс кута дорівнює:

$$\begin{aligned}
L_x &= \iiint \left( y * (\omega_x * y - \omega_y * x) - z * (\omega_z * x - \omega_x * z) \right) * dm = \iiint (\omega_x * \\
&(y^2 + z^2) - \omega_y * x * y - \omega_z * z * x) * dm = \omega_x * \iiint (y^2 + z^2) * dm - \omega_y * \\
&\iiint (x * y) * dm - \omega_z * \iiint (z * x) * dm, \\
\end{aligned}
\tag{3.16}$$

$$\begin{aligned}
L_y &= \iiint \left( z * (\omega_y * y - \omega_z * z) - x * (\omega_x * y - \omega_y * x) \right) * dm = \iiint (\omega_y * \\
&(z^2 + x^2) - \omega_z * z * y - \omega_x * x * y) * dm = \omega_y * \iiint (z^2 + x^2) * dm - \omega_z * \\
&\iiint (z * y) * dm - \omega_x * \iiint (x * y) * dm, \\
\end{aligned}
\tag{3.17}$$

$$\begin{aligned}
L_z &= \iiint \left( x * (\omega_z * x - \omega_x * z) - y * (\omega_y * z - \omega_z * y) \right) * dm = \iiint (\omega_z * \\
&(x^2 + y^2) - \omega_x * x * z - \omega_y * y * z) * dm = \omega_z * \iiint (x^2 + y^2) * dm - \omega_x * \\
&\iiint (x * z) * dm - \omega_y * \iiint (y * z) * dm, \\
\end{aligned}
\tag{3.18}$$

або у векторій формі

$$\vec{L} =$$

$$\begin{pmatrix} L_x \\ L_y \\ L_z \end{pmatrix} = \begin{pmatrix} \iiint (y^2 + z^2) * dm & - \iiint (x * y) * dm & - \iiint (x * z) * dm \\ - \iiint (x * y) * dm & \iiint (z^2 + x^2) * dm & - \iiint (z * y) * dm \\ - \iiint (x * z) * dm & - \iiint (z * y) * dm & \iiint (x^2 + y^2) * dm \end{pmatrix} * \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} =$$

$$= J * \vec{\omega}, [13] \quad (3.19)$$

Де J є так званим тензором інерції (ТоІ) і його компоненти можна переписати наступним чином:

$$J = \begin{pmatrix} \iiint (y^2 + z^2) * dm & - \iiint (x * y) * dm & - \iiint (x * z) * dm \\ - \iiint (x * y) * dm & \iiint (z^2 + x^2) * dm & - \iiint (z * y) * dm \\ - \iiint (x * z) * dm & - \iiint (z * y) * dm & \iiint (x^2 + y^2) * dm \end{pmatrix} =$$

$$= \begin{pmatrix} J_{xx} & J_{xy} & J_{xz} \\ J_{yx} & J_{yy} & J_{yz} \\ J_{zx} & J_{zy} & J_{zz} \end{pmatrix}, [13] \quad (3.20)$$

де перший індекс J співвідноситься з індексом  $\vec{L}$  та другий індекс  $\vec{\omega}$  та  $J_{xy} = J_{yx}$ ;  $J_{xz} = J_{zx}$ ;  $J_{yz} = J_{zy}$ .

Кожен компонент тензора інерції є моментом інерції (МоІ) навколо кожної вісі. Ці компоненти являються постійними, починаючи з того, як початок системи відліку хуz є під'єднаний до тіла. Тензор інерції (ТоІ) може бути спрощений у випадку, якщо осі система відліку хуz збігаються з основними осями тіла (осі симетрії). Для забезпечення цієї можливості для обертового тіла, осі опорної координат повинні бути зафіксовані тілом. Введемо нову систему відліку  $x_b y_b z_b$ , осі якої збігаються з основними осями тіла і знаходиться у нерухомій точці тіло. У додатковій системі відліку ' $x_b y_b z_b$ ' компоненти  $J_{xy} = J_{xz} = J_{yz} = 0$  та рівняння. (3.16) і рівняння. (3.17) можна переписати як:

$$\vec{L} = \begin{pmatrix} L_{xb} \\ L_{yb} \\ L_{zb} \end{pmatrix} = \begin{pmatrix} J_{xbxb} & 0 & 0 \\ 0 & J_{ybzb} & 0 \\ 0 & 0 & J_{zbzb} \end{pmatrix} = \begin{pmatrix} \omega_{xb} \\ \omega_{yb} \\ \omega_{zb} \end{pmatrix} = J * \vec{\omega}, \quad (3.21)$$

Для отримання імпульсу кута використовується рівняння відносного руху:

$$\dot{\vec{r}}_a = \dot{\vec{r}}_b + \vec{\omega} \times \vec{r}, [14] \quad (3.22)$$

де  $\vec{r}_a$  є довільним вектором в інерціальній системі відліку,  $\vec{r}_b$  той самий вектор у неінерційній системі відліку,  $\vec{\omega}$  - кутова швидкість тіла у нерухомій системі відліку. Виведення імпульсу кута  $\frac{d\vec{L}}{dt}$  відповідно до (3.19), призводить до так званого рівняння Ейлера [15]:

$$\vec{\tau} = \frac{d\vec{L}}{dt} = \frac{d(J*\vec{\omega})}{dt} + \vec{\omega} + J*\vec{\omega} = J*\frac{d\vec{\omega}}{dt} + \vec{\omega} + J*\vec{\omega}, [15] \quad (3.23)$$

Рівняння (3.20) може бути переписано більш детально

$$\begin{aligned} \vec{\tau} = \begin{pmatrix} \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} &= \begin{pmatrix} J_{xbxb} & 0 & 0 \\ 0 & J_{ybyb} & 0 \\ 0 & 0 & J_{zbzb} \end{pmatrix} * \begin{pmatrix} \dot{\omega}_{xb} \\ \dot{\omega}_{yb} \\ \dot{\omega}_{zb} \end{pmatrix} + \begin{pmatrix} \omega_{xb} \\ \omega_{yb} \\ \omega_{zb} \end{pmatrix} \\ &\times \begin{pmatrix} J_{xbxb} & 0 & 0 \\ 0 & J_{ybyb} & 0 \\ 0 & 0 & J_{zbzb} \end{pmatrix} * \begin{pmatrix} \dot{\omega}_{xb} \\ \dot{\omega}_{yb} \\ \dot{\omega}_{zb} \end{pmatrix} = \begin{pmatrix} J_{xbxb} * \dot{\omega}_{xb} \\ J_{ybyb} * \dot{\omega}_{yb} \\ J_{zbzb} * \dot{\omega}_{zb} \end{pmatrix} + \begin{pmatrix} \omega_{xb} \\ \omega_{yb} \\ \omega_{zb} \end{pmatrix} \times \\ &\times \begin{pmatrix} J_{xbxb} * \dot{\omega}_{xb} \\ J_{ybyb} * \dot{\omega}_{yb} \\ J_{zbzb} * \dot{\omega}_{zb} \end{pmatrix} = \begin{pmatrix} J_{xbxb} * \dot{\omega}_{xb} \\ J_{ybyb} * \dot{\omega}_{yb} \\ J_{zbzb} * \dot{\omega}_{zb} \end{pmatrix} + \begin{pmatrix} (J_{zbzb} - J_{ybyb}) * \omega_{yb} * \omega_{zb} \\ (J_{xbxb} - J_{zbzb}) * \omega_{xb} * \omega_{zb} \\ (J_{zbzb} - J_{xbxb}) * \omega_{xb} * \omega_{yb} \end{pmatrix}, \quad (3.24) \end{aligned}$$

Або у короткій формі:

$$\begin{aligned} \tau_{xb} &= J_{xbxb} * \dot{\omega}_{xb} + (J_{zbzb} - J_{ybyb}) * \omega_{yb} * \omega_{zb} \\ \tau_{yb} &= J_{ybyb} * \dot{\omega}_{yb} + (J_{xbxb} - J_{zbzb}) * \omega_{xb} * \omega_{zb}, [15] \\ \tau_{zb} &= J_{zbzb} * \dot{\omega}_{zb} + (J_{ybyb} - J_{xbxb}) * \omega_{xb} * \omega_{yb} \end{aligned} \quad (3.25),$$

Рівняння (3.22) слід переписати у формі для знаходження зміни кутової швидкості:

$$\begin{aligned} \dot{\omega}_{xb} &= \frac{\tau_{xb} - (J_{zbzb} - J_{ybyb}) * \omega_{yb} * \omega_{zb}}{J_{xbxb}} \\ \dot{\omega}_{yb} &= \frac{\tau_{yb} - (J_{xbxb} - J_{zbzb}) * \omega_{xb} * \omega_{zb}}{J_{ybyb}}, \end{aligned} \quad (3.26)$$

$$\dot{\omega}_{zb} = \frac{\text{net}\tau_{zb} - (J_{ybyb} - J_{xbxb}) * \omega_{xb} * \omega_{yb}}{J_{zbzb}}$$

Як було сказано вище, опис обертання в системі відліку тіла  $x_b y_b z_b$  замість фіксованої системи відліку хуз спрощує обчислення кутового імпульсу  $\vec{L}$  до рівн. 3.18. З іншого боку, через це спрощення, інше рівняння, які пов'язують кутову швидкість  $\vec{\omega}$  у  $x_b y_b z_b$  і орієнтація  $x_b y_b z_b$  також необхідно визначити відповідно до хуз. Орієнтація тіла визначається унікальним обертанням навколо миттєвої осі обертання. Це обертання можна вважати як сума трьох простих обертань. Послідовності простих обертів не є унікальними. Послідовність, що використовується в аерокосмічних програмах для літаючих об'єктів, - це поворот yaw-pitch-roll (YPR), де кути, напр. зазначається як  $\psi$ ,  $\theta$ ,  $\phi$  відповідно (рисунок 7.8).

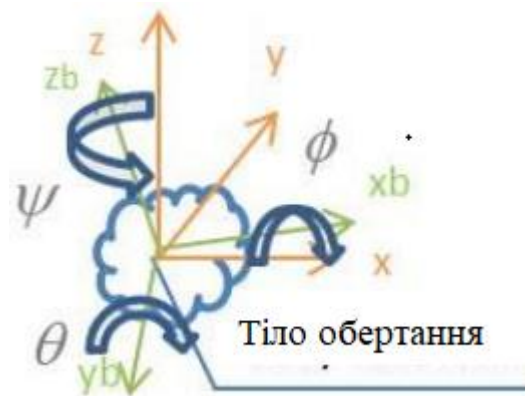


Рисунок 3.10 – Орієнтація тіла обертання  $x_b y_b z_b$  кутами yaw-pitch-roll у фіксованій системі відліку

Зміни орієнтації пов'язані із  $\vec{\omega}$  наступним чином:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin\phi * t\theta & c\phi * t\theta \\ 0 & c\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{c\theta} & \frac{c\phi}{c\theta} \end{pmatrix} * \begin{pmatrix} \omega_{xb} \\ \omega_{yb} \\ \omega_{zb} \end{pmatrix}, \quad [16] \quad (3.27)$$

Таким чином для блоку "6DOF" (з рис. 3.5) потрібні три рівняння: рівняння. (3.12) описує динаміку лінійного руху, рівн. (3.24) описує кінематику кутового руху та рівн. (3.23) описує динаміку кутового руху.

### 3.2.5 Тензор інерції та маси квадрокоптера

Для розрахунку ТоІ, реальна структура квадрокоптера (рис. 3.3) спрощена до структури, яка складається з сферичного щільного центру з масою  $M$ , радіус  $R$  і кілька точкових мас  $m_M$  озташованих на відстані  $l$  [16] (рисунок 3.11).

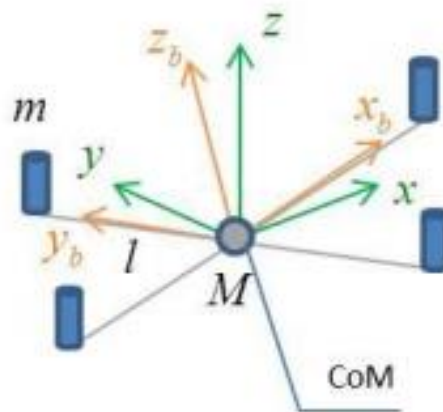


Рисунок 3.11 – Спрощена структура квадрокоптера

ТОІ спрощеної структури на основі рівняння (3.17) та рівняння. (3.18) можна описати як:

$$J = \begin{pmatrix} \iiint (y^2 + z^2) * dm & 0 & 0 \\ 0 & \iiint (z^2 + x^2) * dm & 0 \\ 0 & 0 & \iiint (x^2 + y^2) * dm \end{pmatrix} = \begin{pmatrix} J_{xbxb} & 0 & 0 \\ 0 & J_{ybyb} & 0 \\ 0 & 0 & J_{zbzb} \end{pmatrix}, \quad (3.28)$$

$$\text{Де } J_{xbxb} = J_{ybyb} = \frac{2 * M * R^2}{5} + 2 * l^2 * m_M \text{ та } J_{zbzb} = \frac{2 * M * R^2}{5} + 4 * l^2 * m_M.$$



Рівняння (3.25) доповнює (3.23).

У разі обраного наближення маса квадрокоптера становить:

$$m = M + 4 \cdot m_M \quad (3.29),$$

де  $m_M$  - загальна маса двигуна і лопатей,  $M$  - маса решти частин квадрокоптера.

Рівняння (3.20) доповнює рівняння (3.12).

### 3.2.6 Підсумок рівнянь для руху в 6DOF

Вільний рух квадрокоптера може бути представлений у системі відліку XYZ (рисунок 3.12)

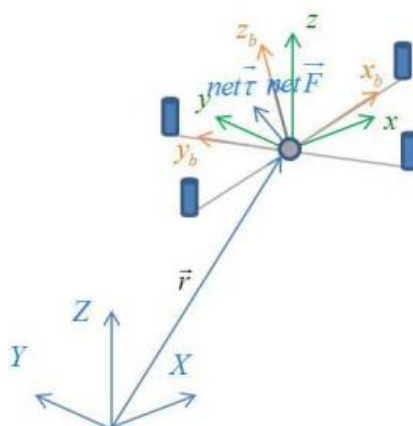


Рисунок 3.12 – Вільний рух квадрокоптера

Параметри та функції цього руху можна визначити в кілька етапів. Початкові умови встановлюються відповідно до поточного експерименту (наприклад, дорівнює нулю). Потім маса і ToI обчислюються на основі рівняння (3.25) та рівняння (3.26). Згодом динаміка та кінематика орієнтації розраховується на основі рівняння (3.24) та рівняння (3.23) та динаміка лінійного руху обчислюється на основі рівняння (3.12).

### 3.2.7 Зовнішні сили

Положення та орієнтація квадрокоптера можна знайти за допомогою рівняння (3.12) та рівн. (3.23) якщо зовнішня сила  $\text{net}\vec{F}$  і зовнішній крутний момент  $\text{net}\vec{\tau}$  відомі.

Як було сказано раніше, існує три джерела зовнішніх сил: гравітаційне поле, опір повітря та обертання лопатей у повітрі, що призводять до гравітаційної сили  $\vec{F}_{mg}$ , сила тяги  $\vec{F}_{drag}$ , і сила тяги  $\vec{T}$  відповідно. Силами та моментами лез можна керувати. Гравітаційною силою та силою опору повітря контролювати неможливо.

#### 3.2.7.1 Гравітаційні сили

Сили гравітаційного поля, прикладені до тіла, можна представити у вигляді рівнодійної гравітаційної сили, що прикладена до центру ваги (CoG) тіла. Напрямок цієї сили постійний і вказує на центр Землі. Зв'язок між цими силами і прискоренням об'єкта відповідає закону Ньютона і його можна записати як:

$$\vec{F}_{mg} = m * \vec{g}, \quad (3.30)$$

де  $m$  - маса квадрокоптера та  $\vec{g}$  - гравітаційне прискорення.

Для спрощення припустимо, що центр ваги (CoG) збігається з центром мас квадрокоптеру (CoM.)

#### 3.2.7.2 Опір повітря

Коли предмет рухається у повітрі, він зазнає опору до повітря. Повітряна сила тяги може бути описана як:

$$F_{drag} = C_d * \rho * S * v^2, \quad [17] \quad (3.31),$$

де  $C_d$  це коефіцієнт аеродинамічного опору,  $\rho$  - масова щільність повітряної рідини,  $S$  розрахункова площа об'єкта,  $v$  - швидкість об'єкта відносно повітряної рідини.

Коефіцієнт  $C_d$ , який залежить від форми об'єкта, слід заздалегідь виміряти у аеродинамічній трубі. Коефіцієнти аеродинамічного опору для декількох фігур добре відомі і доступні у вигляді таблиць, наприклад для квадратної форми  $C_d$  дорівнює 0,64 [18]. Масова щільність повітря  $\rho$  залежить від висоти над рівнем моря. Наприклад, на 1 метр над рівнем моря з температурою близько 15 градусів щільність повітря дорівнює 1,226 [18].

Для руху у 3 вимірах рівн. (3.28) можна переписати як:

$$\vec{F}_{drag} = C_d * \rho * S * v / v * \vec{v}, \quad (3.32)$$

### 3.2.8 Сили від лез

Взаємодія між обертовим лезом і повітрям може бути описана теорією вихору [18]. Припустимо, що лезо обертається з деякою кутовою швидкістю  $\omega_{bl}$  у напрямку проти годинникової стрілки. Це обертання призводить до створення ряду сил. Щоб знайти рівнодійну сил, поверхня леза теоретично розділена на дрібні елементи; сила, яка прикладена до елемента, представленою як сума вертикальної сили  $\vec{T}_{el}$  та горизонтальної сили  $\vec{Q}_{el}$  (рисунок 3.13). Сума всіх вертикальних сил  $\vec{T}_{el}$  може бути замінена силою тяги  $\vec{T}$  і сума всіх горизонтальних сил  $\vec{Q}_{el}$  на крутний момент  $\vec{H}$ .

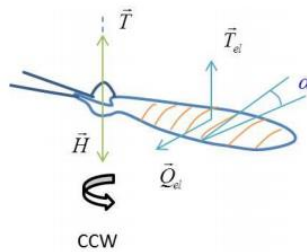


Рисунок 3.13 – Рівнодійна сил і крутний момент від взаємодії між клинком і повітрям

Сила тяги  $\vec{T}$  і крутний момент  $\vec{H}$  можна описати як:

$$\vec{T} = b_T \cdot \omega_{bl}^2, [18] \quad (3.33)$$

$$\vec{H} = b_H \cdot \omega_{bl}^2, [18] \quad (3.34)$$

де  $b_T$  та  $b_H$  це пропорційні коефіцієнти, що залежать від щільності повітря, кута лопаті та площі леза.

Квадрокоптер має чотири мотора; кожен з них складається з лопаті, двигуна та силового мосту. Зазначимо лінійний рух квадрокоптера вперед, назад, вліво і вправо та номери моторів від 1 до 4 (рисунок 3.15). Лопаті 2 і 4 обертаються в напрямку проти годинникової стрілки з кутовою швидкістю  $\omega_2, \omega_4$ , тоді як лопаті 1 і 3 обертаються за годинниковою стрілкою зі швидкістю кута  $\omega_1, \omega_3$ .

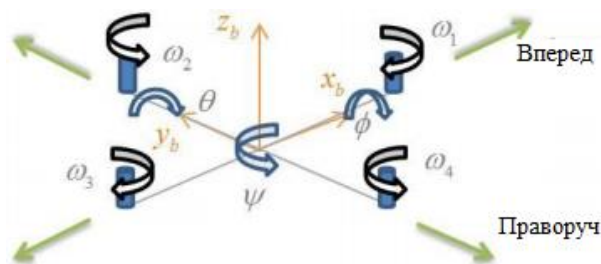


Рисунок 3.14 – Структура квадрокоптера

Ці обертання призводять до чотирьох пар сил тяги та крутних моментів (рисунок 3.15).

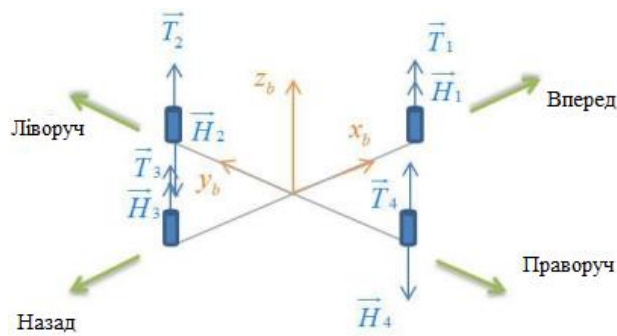


Рисунок 3.15 – Сила тяги з кожного леза

Ці 4 сили можна замінити на рівнодійну сил тяги  $\vec{T}$  та 4 крутячих моментів

$\vec{H}$  (рисунок 3.15).

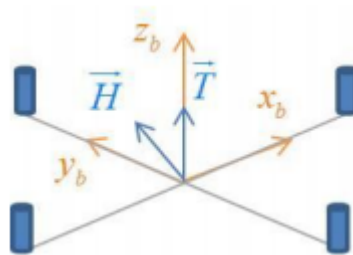


Рисунок 3.16 – Сила тяги та крутний момент застосоввані до квадрокоптера  
(обрано довільний напрямок  $\vec{H}$ )

Сила тяги  $\vec{T}$  може бути представлено у вигляді:

$$\vec{T} = \vec{T}_1 + \vec{T}_2 + \vec{T}_3 + \vec{T}_4, [18] \quad (3.35).$$

Крутний момент  $\vec{H}$  може бути представлено у вигляді:

$$\vec{H} = \begin{pmatrix} H_{xb} \\ H_{yb} \\ H_{zb} \end{pmatrix} = \begin{pmatrix} l * (\vec{T}_2 - \vec{T}_4) \\ l * (\vec{T}_3 - \vec{T}_1) \\ (\vec{H}_1 + \vec{H}_3) - (\vec{H}_2 + \vec{H}_4) \end{pmatrix}, \quad (3.36), [18]$$

де знак мінус відповідає негативному напрямку кутів крену, тангажу та ролу.

Рівняння (3.33) доповнює рівн. (3.23) для розрахунку орієнтації квадрокоптеру, тому (3.23) можна переписати як:

$$\begin{aligned}\dot{\omega}_{xb} &= \frac{l * (\vec{T}_2 - \vec{T}_4) - (J_{zbzb} - J_{ybzb}) * \omega_{yb} * \omega_{zb}}{J_{xbxb}} \\ \dot{\omega}_{yb} &= \frac{l * (\vec{T}_3 - \vec{T}_1) - (J_{xbxb} - J_{zbzb}) * \omega_{xb} * \omega_{zb}}{J_{ybyb}}, \\ \dot{\omega}_{zb} &= \frac{(\vec{H}_1 + \vec{H}_3) - (\vec{H}_1 + \vec{H}_3) - (J_{ybyb} - J_{xbxb}) * \omega_{xb} * \omega_{yb}}{J_{zbzb}}\end{aligned}\quad (3.28)$$

Для знаходження передачі сили тяги  $\vec{T}$  повинні бути представлені в XYZ як:

$$\vec{T} = \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} = R_{xyz}^{x_b y_b z_b} * \vec{T}_{x_b y_b z_b}, \quad (3.29)$$

де  $R_{xyz}^{x_b y_b z_b}$  це матриця обертання.

Матриця обертання YPR тоді буде:

$$\begin{aligned}R_{xyz}^{x_b y_b z_b} &= [R_x(\varphi) * R_y(\theta) * R_z(\psi)]^{-1} = \\ &= \left[ \begin{pmatrix} 1 & 0 & 0 \\ 0 & c\varphi & sn\varphi \\ 0 & -sn\varphi & c\varphi \end{pmatrix} * \begin{pmatrix} c\theta & 0 & -sn\theta \\ 0 & 1 & 0 \\ sn\theta & 0 & c\theta \end{pmatrix} * \begin{pmatrix} c\psi & sn\psi & 0 \\ -sn\psi & c\psi & 0 \\ 0 & -0 & 1 \end{pmatrix} \right]^{-1} = \\ &= \left[ \begin{pmatrix} c\theta * c\psi & c\theta * sn\psi & -sn\theta \\ sn\varphi * sn\theta * c\psi - c\varphi * sn\psi & sn\varphi * sn\theta * sn\psi - c\varphi * c\psi & sn\varphi * c\theta \\ c\varphi * sn\theta * c\psi - sn\varphi * sn\psi & c\varphi * sn\theta * sn\psi - sn\varphi * c\psi & c\varphi * c\theta \end{pmatrix} \right]^{-1} = \\ &= \begin{pmatrix} c\theta * c\psi - sn\varphi * sn\theta * sn\psi & -c\theta * sn\psi & c\psi * sn\theta + c\theta * sn\varphi * sn\psi \\ c\varphi * sn\psi + sn\varphi * sn\theta * c\psi & c\varphi * c\psi & sn\theta * sn\psi - sn\varphi * c\theta * c\psi \\ -c\theta sn\theta & sn\varphi & c\varphi * c\theta \end{pmatrix}, \quad (3.30)\end{aligned}$$

де c та sn це скорочення для функцій sin та cos відповідно.

$$\begin{aligned}\vec{T}_{xyz} &= \begin{pmatrix} T_x \\ T_y \\ T_z \end{pmatrix} = R_{xyz}^{x_b y_b z_b} * \vec{T} = R_{xyz}^{x_b y_b z_b} * \begin{pmatrix} 0 \\ 0 \\ T \end{pmatrix} = \\ &= \begin{pmatrix} c\theta * c\psi - sn\varphi * sn\theta * sn\psi & -c\theta * sn\psi & c\psi * sn\theta + c\theta * sn\varphi * sn\psi \\ c\varphi * sn\psi + sn\varphi * sn\theta * c\psi & c\varphi * c\psi & sn\theta * sn\psi - sn\varphi * c\theta * c\psi \\ -c\theta sn\theta & sn\varphi & c\varphi * c\theta \end{pmatrix} * \begin{pmatrix} 0 \\ 0 \\ T \end{pmatrix} \\ &= \begin{pmatrix} c\psi * sn\theta + sn\varphi * sn\psi \\ sn\theta * sn\psi - sn\varphi * c\theta * c\psi \\ c\varphi * c\theta \end{pmatrix} * T, \quad (3.31)\end{aligned}$$

З урахуванням рівн. (3.37), гравітаційна сила рівн. (3.27) і сила тяги рівн. (3.29), рівн. (3.12) можна переписати як:

$$\begin{aligned}\ddot{r}_x &= \frac{(c\psi * sn\theta + c\theta * sn\varphi * sn\psi) * T * C_d * \rho * S * v_x^2}{m} \\ \ddot{r}_y &= \frac{(sn\theta * sn\psi - sn\varphi * c\theta * c\psi) * T * C_d * \rho * S * v_y^2}{m}, \quad (3.32). \\ \ddot{r}_z &= \frac{c\varphi * c\theta - m * g - C_d * \rho * S * v_z^2}{m}\end{aligned}$$

Таким чином, вільний рух квадрокоптера в 6DOF можна представити як лінійний рух його центр маси (ЦМ) з масою  $m$  в  $XYZ$  і кутове обертання квадрокоптера навколо центру мас. Обертальних рух описується обертанням квадрокоптера в  $x_b y_b z_b$  та орієнтацією  $x_b y_b z_b$  відносно  $x y z$ , де осі  $x y z$  паралельні відносним осям  $XYZ$  (рисунок 3.17).

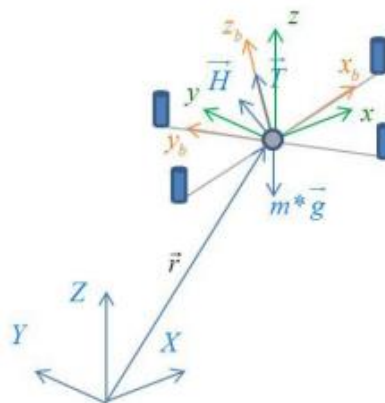


Рисунок 3.17 – Рух квадрокоптера у 6DOF

Вплив прикладених сил описується рівнянням. (3.38) (замість (3.12)) та вплив застосовуваних крутних моментів описується рівнянням. (3.34) (замість 3.23).

### 3.2.9 Приводи квадрокоптера.

Привід складається з лопаті, двигуна та моста живлення. Поведінка леза описана в попередньому розділі рівняннями (3.30), рівн. (3.31).

Кутові швидкості лопатей залежать від кутових швидкостей відповідних двигунів. Для розрахунку швидкості вала безщіткового двигуна постійного струму привода слід використовувати математичну модель двигуна. Ця модель залежить від конструкції двигунів. На відміну від щіткового двигуна постійного струму, двигуну привода необхідна система управління для обертання його ротора. Іноді ММ привода потрібна для створення такого типу системи управління. Однак для поточного випадку ММ необхідний для оцінки співвідношення між входом та виходом. Для цього ММ привода можна замінити ММ щіткового постійного струму [19]. ММ щіткового постійного струму описуються за допомогою чотирьох рівнянь:

$$U = i * R + L * \frac{di}{dt} + e_m, \quad (3.33)$$

$$E_m = k_\omega * \omega, \quad (3.34)$$

$$T = k_\omega * i, \quad (3.35)$$

$$T = J * \frac{d\omega}{dt}, \quad (3.36).$$

Рівняння (3.39) описує ефект, коли прикладена напруга призводить до струму в арматурі з опором  $R$  і індуктивності  $L$  і до повернення ЕРС  $e_m$ . Рівняння (3.40) вказує на те, що ЕДС заднього моста  $e_m$  пропорційна кутовій



швидкості валу двигуна, де  $k_{\omega}$  це константа ЕДС заднього моста. Рівняння (3.41) позначає, що вироблений крутний момент пропорційний виробленому струму, де  $k_m$  - константа крутного моменту. Рівняння (3.42) описує перехід від крутного моменту до кутового прискорення установки, де  $J$  - сума моменту інерції установки та валу двигуна. Для поточного випадку, установка - це лезо, яке має мінімальний момент інерції, тому момент інерції установи можна опустити.

Силовий міст може бути представлений у вигляді:

$$u = k_{PWM} * u_{max}, \quad (3.37)$$

де  $u$  - напруга на вході до мотору,  $u_{max}$  - максимальна вхідна напруга мотору, що подається на силовий міст, а  $k_{PWM}$  - відсоток ширини імпульсної модуляції (ШІМ). Часовою затримкою силового мосту можна знехтувати, оскільки час для зміни електричних сигналів значно менший порівняно із затримкою часу в механічній частині системи. Силовий міст можна розглядати як безперервний елемент.

### 3.2.10 Дискретні елементи

Система має кілька дискретних елементів, таких як: мікроконтролер та датчики. Ці елементи дискретні за часом та рівнем. Вони повинні бути замінені кванторами з рівнем дискретизації  $ld_M$  і  $ld_s$ , що відповідають точності їх обчислення, а дискретизація часу  $td_M$  і  $td_s$  відповідає їх затримці у часі.

Ці параметри можна обчислити як:

$$ld_M = \frac{1}{2^{NmofBt}} ; ld_s = \frac{1}{2^{NmofBt}}, \quad (3.38)$$

де  $NmofBt$  - довжина регістра мікроконтролера у байтах.

$$td_s = \frac{1}{f}, \quad (3.39)$$

де  $f$  - частота датчика в Гц.

$$td_M = N * t_{ins}, \quad (3.40)$$

де  $N$  - це кількість інструкцій у коді та  $t_{ins}$  - час для виконання однієї інструкції.

### 3.2.11 Математична модель квадрокоптера

ММ квадрокоптера складається з передавальної функції (TF) елементів, що описані в попередніх розділах. Логічна схема цієї моделі з відповідною рівняння показано на рисунку 3.18.

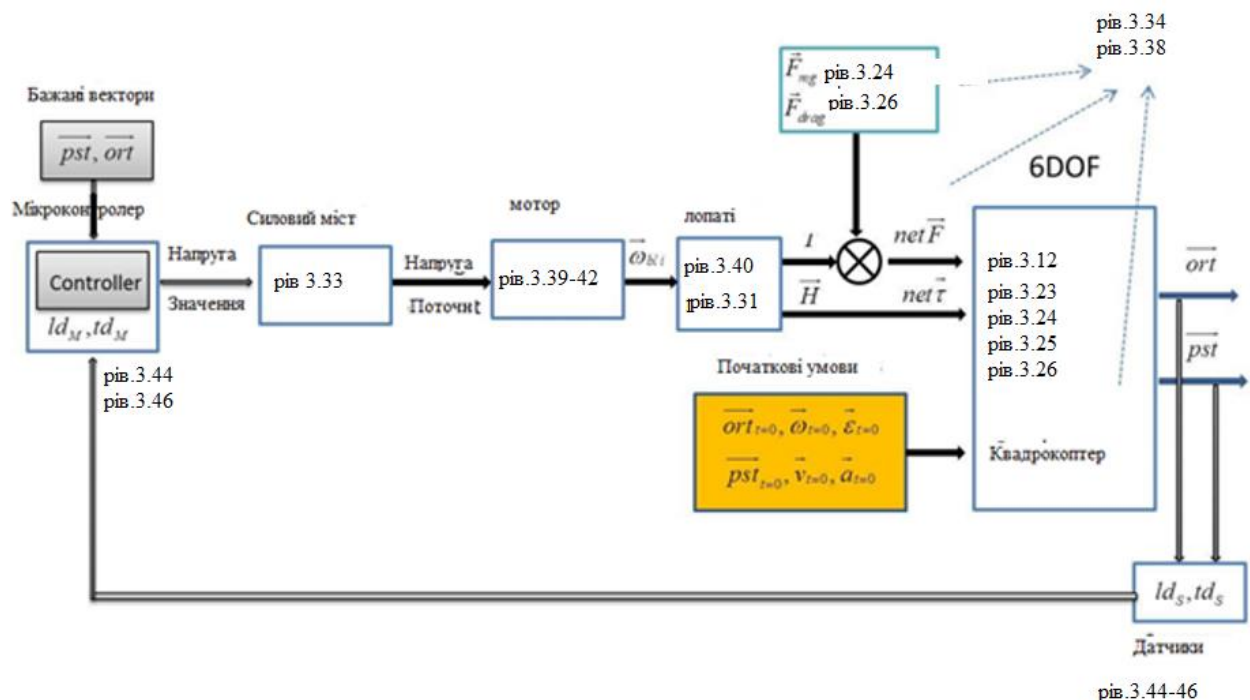


Рисунок 3.18 – Логічна схема ММ квадрокоптера в 6DOF

Ця модель має такі спрощення:

- квадрокоптер - це тверде тіло
- центр маси (ЦМ) знаходиться в геометричному центрі квадрокоптера
- ТоІ квадрокоптера апроксимується як інерційний момент декількох об'єктів

- Центр тяжіння збігається з центром мас
- Затримка мосту в часі нехтується

Для цілей моделювання рівняння повинні бути представлені функціями передачі. TF буде представлена у формі, придатній для реалізації в Matlab / Simulink. TF для рівн. (3.38):

$$r_x(s) = \frac{1}{s^2} \left( \frac{(c\psi(s)*sn\theta(s) + c\theta(s)*sn\varphi(s)*sn\psi(s))}{m} * T(s) + \frac{-Cd*\rho*S}{m} * (sr_x(s))^2 \right)$$

$$r_y(s) = \frac{1}{s^2} \left( \frac{sn\theta(s)*sn\psi(s) - sn\varphi(s)*c\theta(s)*c\psi(s)}{m} * T(s) + \frac{-Cd*\rho*S}{m} * (sr_y(s))^2 \right), \quad (3.41)$$

$$r_z(s) = \frac{1}{s^2} \left( \frac{c\varphi(s)*c\theta(s)}{m} * T(s) + \frac{-Cd*\rho*S}{m} * (sr_z(s))^2 \right)$$

де  $s$  – оператор Лапласу.

Ця передавальна функція нелінійна, оскільки вона має тригонометричні функції та квадратування входів.

На основі рівн. (3.41) він може бути представлений у Matlab / Simulink (рисунок 3.19)

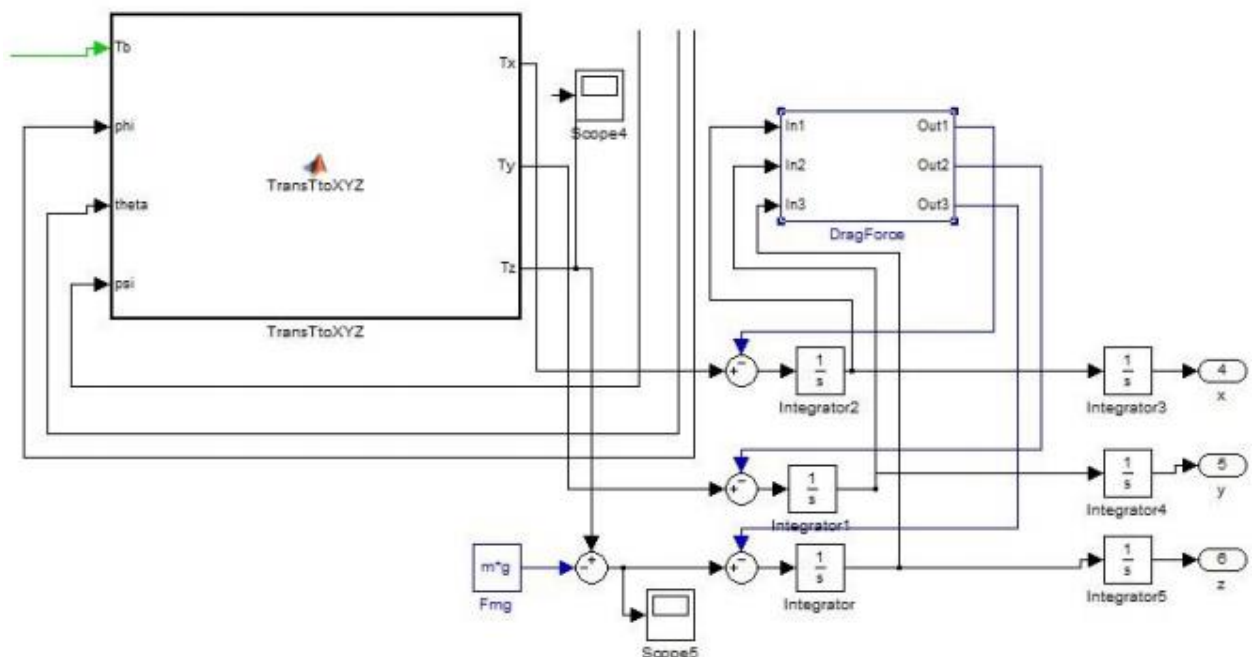


Рисунок 3.19 – Представлення передавальної функції для передачі руху у Matlab/Simulink

Блок "TransTtoXYZ" (рисунок 3.19) має як вхід сили тяги  $T(s)$  так і кути орієнтації  $\theta, \psi, \varphi$ . Ці входи повинні бути обчислені для пошуку положення квадрокоптера. Передавальні функції для кутів орієнтації розраховуються на основі рівняння. (3.24) та рівняння. (3.34) як:

$$s \begin{pmatrix} \varphi(s) \\ \theta(s) \\ \psi(s) \end{pmatrix} = \begin{pmatrix} 1 & sn\varphi(s) * t\theta(s) & c\varphi(s) * t\theta(s) \\ 0 & c\varphi(s) & -sn\varphi(s) \\ 0 & \frac{sn\varphi(s)}{c\theta(s)} & \frac{c\varphi(s)}{c\theta(s)} \end{pmatrix} * \begin{pmatrix} \omega_{x_b}(s) \\ \omega_{y_b}(s) \\ \omega_{z_b}(s) \end{pmatrix}, \quad (3.42)$$

$$s\omega_{x_b}(s) = \frac{l}{J_{x_b x_b}} * (\vec{T}_2(s) - \vec{T}_4(s)) + \frac{-(J_{z_b z_b} - J_{y_b y_b})}{J_{x_b x_b}} * \omega_{y_b}(s) * \omega_{z_b}(s)$$

$$s\omega_{y_b}(s) = \frac{l}{J_{y_b y_b}} * (\vec{T}_3(s) - \vec{T}_1(s)) + \frac{-(J_{x_b x_b} - J_{z_b z_b})}{J_{y_b y_b}} * \omega_{x_b}(s) * \omega_{z_b}(s), \quad (3.43).$$

$$s\omega_{z_b}(s) = \frac{l}{J_{z_b z_b}} * (H_1(s) + H_3(s)) - (H_2(s) + H_4(s)) \frac{-(J_{y_b y_b} - J_{x_b x_b})}{J_{z_b z_b}} * \omega_{x_b}(s) * \omega_{y_b}(s)$$

Ці передавальні функції нелінійні, оскільки вони мають тригонометричні функції та множення результатів. Вони можуть бути представлені в Matlab / Simulink, як показано на Рисунку 3.20.

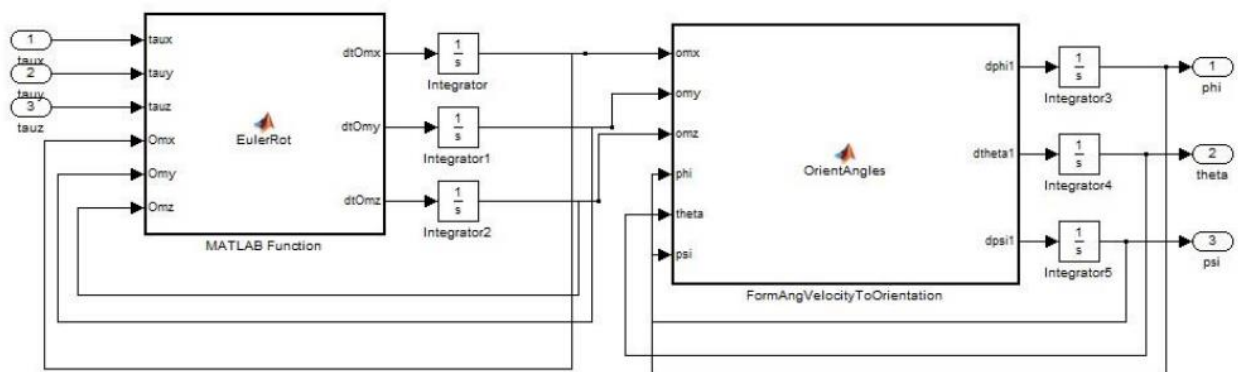


Рисунок 3.20 – Вхід цього блоку - крутний момент  $H(s)$ . Входи  $T(s)$  і  $H(s)$  можна обчислити, виходячи із рівнянь, (3.40), (3.41), (3.42), (3.43).

Передавальні функції:

Зм	Лист	№ докум.	Підп.	Дата

$$T_1(s)=b_T*\omega_{bl1}^2(s); T_2(s)=b_T*\omega_{bl2}^2(s); T_3(s)=b_T*\omega_{bl3}^2(s); T_4(s)=b_T*\omega_{bl4}^2(s), \quad (3.44)$$

$$H_1(s)=b_H*\omega_{bl1}^2(s); H_2(s)=b_H*\omega_{bl2}^2(s); H_3(s)=b_H*\omega_{bl3}^2(s); H_4(s)=b_H*\omega_{bl4}^2(s), \quad (3.45)$$

$$T(s)= b_T*( \omega_{bl1}^2(s)+ \omega_{bl2}^2(s)+ \omega_{bl3}^2(s)+ \omega_{bl4}^2(s)), \quad (3.46)$$

$$\begin{pmatrix} H_{x_b}(s) \\ H_{y_b}(s) \\ H_{z_b}(s) \end{pmatrix} = \begin{pmatrix} l * (\vec{T}_2(s) - \vec{T}_4(s)) \\ l * (\vec{T}_3(s) - \vec{T}_1(s)) \\ (H_1(s) + H_3(s)) - (H_2(s) + H_4(s)) \end{pmatrix}, \quad (3.47)$$

Ці передавальні функції нелінійні, оскільки вони мають квадратичні результати. Вони можуть бути представлені в Matlab / Simulink(рисунок 3.21)

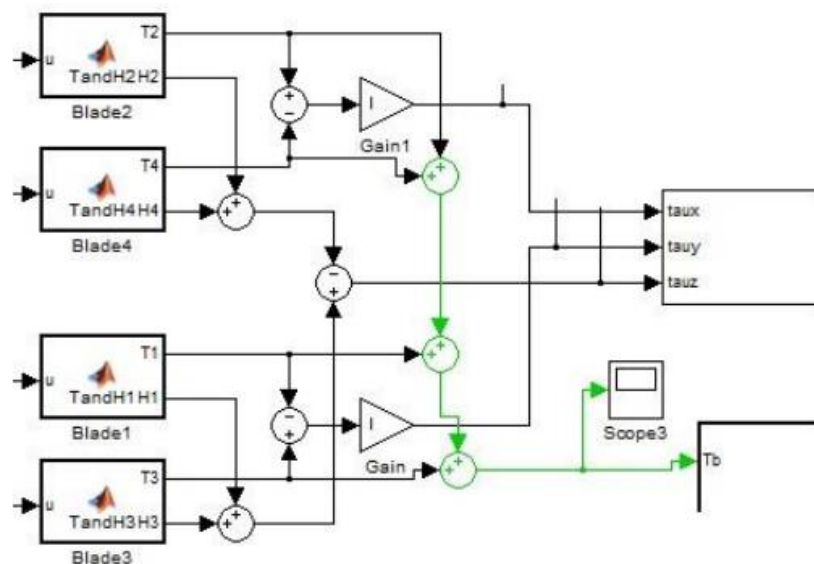


Рисунок 3.21 – Предсавлення передавальної функції у Matlab/Simulink для створення T(s) та H(s).

Математичну модель BLDC розроблено на основі рівн. (3.39 - 40). Застосування перетворення Лапласа в рівн. (3.39), (3.40) призводить до:

$$\frac{U(s)-k_\omega*\omega(s)}{R+sL}=I(s), \quad (3.48)$$

та до рівняння (3.41), (3.42) призводить до:

$$k_m * I(s)=J*s\omega(s). \quad (3.49)$$

Передавальну функцію можна отримати, комбінуючи рівн. (3.54) та(3.55) як:

$$k_m * \frac{U(s) - k_\omega * \omega(s)}{R + sL} = J * s \omega(s) \Rightarrow U(s) \frac{J}{k_m} * (R + sL) * s \omega(s) + k_\omega * \omega(s) \Rightarrow$$

$$U(s) = \frac{1}{\frac{J * L}{k_m} s^2 + \frac{J * R}{k_m} s + k_\omega} = \omega(s), (3.50)$$

де  $G(s) = \frac{1}{\frac{J * L}{k_m} s^2 + \frac{J * R}{k_m} s + k_\omega}$  це передавальна функція другого порядку.

Більш загальна форма рівняння передавальної функції, де параметри  $J$ ,  $L$  і  $k_m$  заміщені, представлена у більш відповідній формі як:

$$G(s) = \frac{1}{\frac{J * L}{k_m} s^2 + \frac{J * R}{k_m} s + k_\omega} = \frac{\frac{1}{k_\omega}}{\frac{J * R}{k_m * k_\omega} * \frac{J}{R} s^2 + \frac{J * R}{k_m * k_\omega} s + 1} = \frac{\frac{1}{k_\omega}}{\tau_m * \tau_e s^2 + \tau_m s + 1} (3.51)$$

де  $\tau_m = \frac{J * R}{k_m * k_\omega}$  це механічна константа та  $\tau_e = \frac{J}{R}$  – електрична константа.

Запровадження передавальної функції, оснований на рівн.( 3.51) показано на Рисунку 3.22

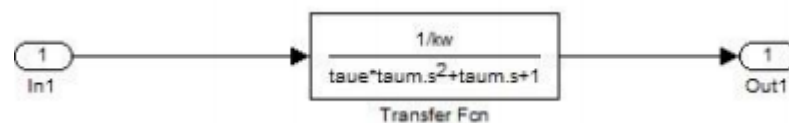


Рисунок 3.22 – Представлення передавальні функції BLDC у Simulink

## Висновки

1. Таким чином було розглянуто принципи ройового інтелекту в якості оптимального рішення для виконання загальних для групи об'єктів завдань.

2. Було розроблено математичну модель (ММ) об'єкта дослідження відповідно до умов, наближених до реальних. Кінцеві результати по моделюванню було представлено у вигляді функціональних схем у MatLab/Simulink.

## 4 РОЗРОБКА АЛГОРИТМІВ УПРАВЛІННЯ РОЄМ ОБ'ЄКТІВ

### 4.1 Алгоритм управління роєм динамічних об'єктів

Оскільки зазвичай визначають як групу поведінкових сутностей які разом узгоджують для отримання значного або бажаного результату чи поведінки, вдаємося до природних прикладів ройової поведінки в природі. Бджоли узгоджують одна одну виконувати завдання, важливі для виживання свого рою. Зграї міграції гусей координують ефективні схеми польоту для досягнення їх міграція. Мурахи виконують загально одну й ту саму задачу поодинці.

Пропонується використовувати наступний узагальнений алгоритм управління роєм агентів на основі консенсусу [20]:

1.Всі агенти, що входять до складу рою, отримують з деякого пункту управління завдання, що стоїть перед роєм.

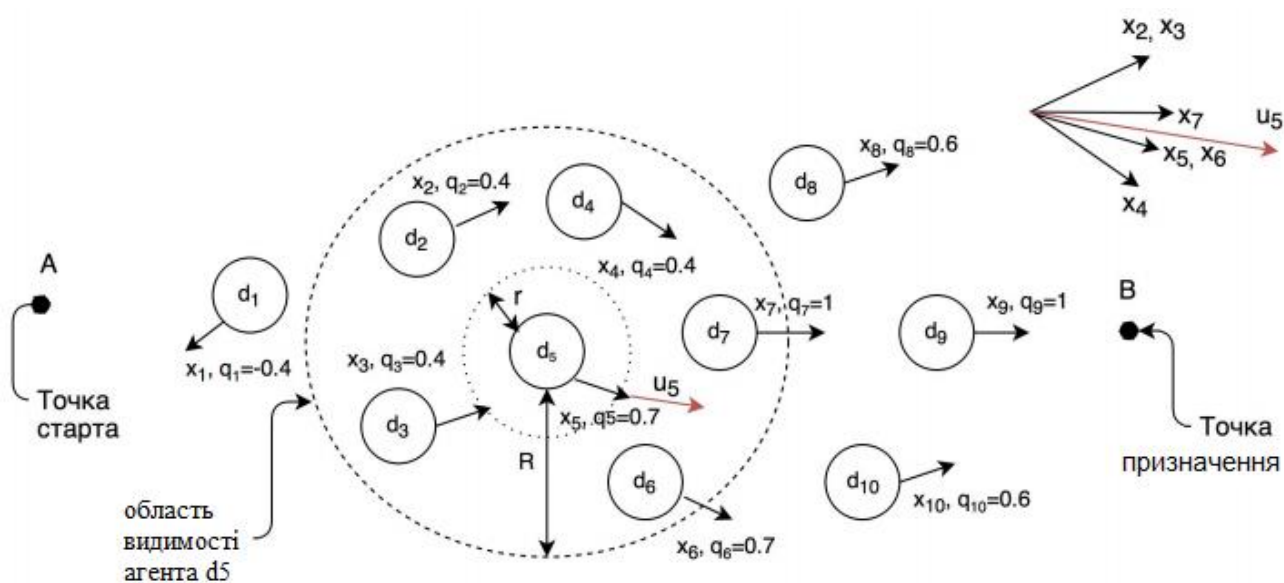


Рисунок 4.1 – Розрахунок зміни траєкторії агентом  $d_5$

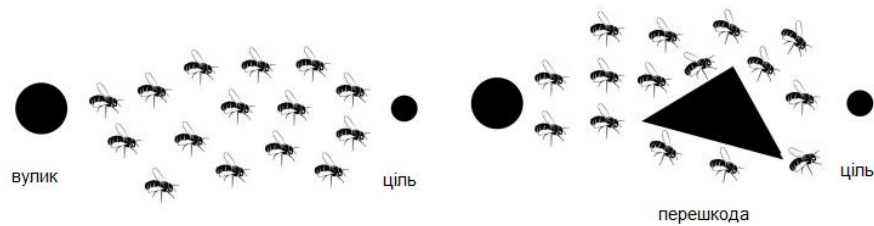


Рисунок 4.2 – Рух рою бджіл без перешкод та з ними.

2. Ініціалізація протоколу голосування, установка зв'язків між сусідами, початок обміну повідомленнями.
  3. Кожен агент  $d_i \in D$  отримує інформацію про поточний напрямку руху об'єктів рою, що знаходяться в зоні його видимості, і числову характеристику потенціалу їх руху.
  4. На підставі власних і отриманих даних, формується управління.
  5. Зміна напрямку руху із застосуванням функції уникнення зіткнень.
  6. Перерахунок потенціалу руху  $q_t$  на основі апріорної інформації агента і додаткових відомостей, одержуваних при дослідженні світу і від сусідів.
- Так як ройовий інтелект засновний на принципах самоорганізації, наведемо алгоритм управління тільки окремою особою рою [21] (алгоритм. 1).

#### procedure SWARMCONTROL

початок виконання,  $t=0$ ;

установка кроку протоколу  $\alpha$ ;

**repeat**

GETDIRECTIONSANDPOTENTIALS;

підрахунок значень  $y_{t,i}$  і  $y_{t,j}$ ;

формування управління  $u_t$ ;

Зм	Лист	№ докум.	Підп.	Дата



COLLISIONAVOIDANCE;

Визначення напрямку руху  $x_{t+1}$

перерахування потенціалу руху  $q_t$ ;

$t = t + 1$ ;

**until** рішення задачі не досягнуто;

**end procedure**

Отримання напрямків руху сусідів і їх потенціалів в асинхронному режимі

**procedure**

GETDIRECTIONSANDPOTENTIALS

**for**  $i = 0; i < \text{size}(\text{self.neighbourList}); i++$  **do**

$\text{self.directions}[i] = \text{self.neighbourList}[i].\text{RequestDirection}();$

$\text{self.potentials}[i] = \text{self.neighbourList}[i].\text{RequestPotential}();$

**end for**

**end procedure**

Алгоритм 1 – Управління роєм динамічних об'єктів на основі консенсуса

## 4.2 Стратегії групового управління

Стратегії групового управління можна розділити на централізовані і децентралізовані. Головною відмінністю централізованих стратегій є наявність центрального керуючого пристрою (ЦКП). Цей пристрій здійснює тотальний контроль над усіма іншими агентами і є головним елементом системи [22]. Перевагою даного підходу є простота інтеграції з алгоритмічної сторони. Серед недоліків можна виділити завантаженість ЦКП, яка буде значно збільшуватися при відносно малому прирості кількості агентів в групі, а також низьку життєздатність.

У свою чергу, децентралізовані стратегії мають велику життєздатність за рахунок відсутності ЦКП. Їх поділяють на колективну стратегію, в якій агенти обмінюються інформацією для досягнення групової мети, і стабільну стратегію, агенти якої функціонують незалежно один від одного.

Перевагою колективного підходу є висока життєздатність системи та її швидкодія.

Недоліками є складність алгоритмізації, а також наявність каналу обміну інформацією, яке може привести до порушення функціонування системи при втручанні деструктивного характеру.

У свою чергу, система, заснована на стабільній стратегії, за рахунок відсутності каналу обміну інформацією має найвищу життєздатність серед інших стратегій. Також серед варто відзначити мінімальну завантаженість агентів, що не залежить від їх кількості, і простоту алгоритмізації [23]. Однак стабільна стратегія не може бути застосована до рою БПЛА для досягнення спільної мети, так як не дозволяє складним елементам БПЛА здійснювати комунікацію, що може привести до порушень у функціонуванні системи.

Отже, використання колективної стратегії управління дозволяє не тільки видалити з системи ЦКП, а й забезпечити оптимальне виконання складних складових завдань групою БПЛА.

#### 4.2.1 Узагальнена модель інформаційної взаємодії.

Від вибору каналу зв'язку залежить не тільки кількість інформації, яку можна передати від передавача до одержувача повідомлень, а й швидкість передачі інформації та її вірогідність [24]. ІВ поділяють на внутрішнє і зовнішнє. У внутрішньому ІВ беруть участь наступні елементи: сенсори і датчики, що збирають інформацію про стан в просторі і перешкоди в навколишньому середовищі; процесорний пристрій (ПП), що відповідає за обробку інформації та формулюванні команд; мотори і інші пристрої,

необхідні для виконання цілей агента. Таким чином, в процесі внутрішнього ІВ передається інформація про координати, положення в просторі (кути тангажа, крену, ризику), перешкодах, технічний стан агента, команди для регулювання положення в просторі і для виконання інших завдань [25].

Криптографічний захист даних в цифрових інформаційних системах вимагає все більшої уваги сучасного суспільства. Сучасне суспільство все більшою мірою спирається на Інформаційні процеси (ІП), які стають рушійною силою економіки, суспільних відносин, військової справи. ІП - це процеси збору, підготовки, передачі, обробки, перетворення та використання інформації в різних сферах суспільства. Саме тому управління різними технічними об'єктами, у тому числі і групами дронів, вимагає впровадження засобів захисту команд управління та потоку даних про стан і траєкторію руху таких об'єктів [25]. БПЛА розглядають як неподільний об'єкт. Інформація, що отримується агентами в процесі обміну інформацією один з одним, підрозділяється на інформацію про географічне розташування інших БПЛА групи, розташування перешкод у середовищі та технічний стан агентів. На основі вищеприписаної інформації система має можливість колективно розподіляти завдання між агентами [25]. Дане дослідження передбачає, що колаборація БПЛА є мережею, стійкої до розривів, отже, всі інформаційні повідомлення доходять до адресата .



Рисунок 4.3 – Модель послідовної передачі інформаційних повідомлень в групі безпілотних літальних апаратів

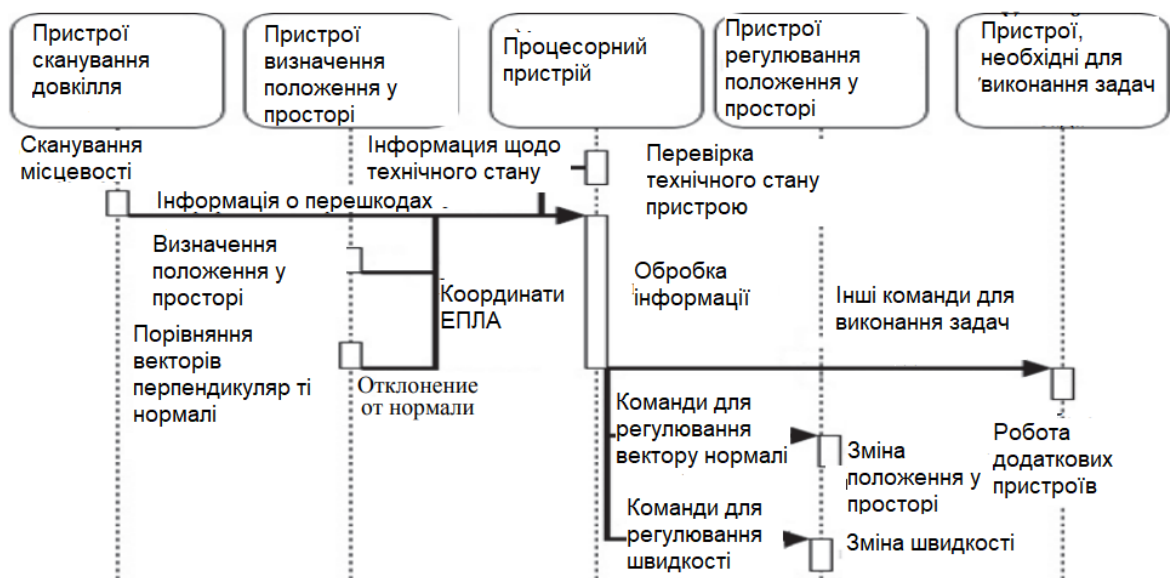


Рисунок 4.4 – Уразливі інформаційні повідомлення в моделі внутрішнього інформаційного взаємодії елементів в групі безпілотних літальних апаратів

#### 4.2.2 Аналіз моделі внутрішнього інформаційного взаємодії.

Аналізуючи внутрішнє ІВ, можна говорити про вразливість всіх елементів, представлених в моделі [26]. Такий висновок можна зробити, помітивши, що пристрої сканування навколишнього середовища і визначення положення в просторі отримують інформацію від навколишнього середовища, що має на увазі можливість її спотворення в процесі зчитування, отже, ці пристрої уразливі для ДІВ(диструктивна ІВ). Після збору необхідних даних пристроєм сканування навколишнього середовища і визначення положення в просторі обробляють і передають ці дані ПП, отже, воно вразливе до ДІВ з боку даних пристроїв [26].

Виявимо уразливості в узагальненій моделі зовнішнього ІВ. Група БПЛА складається з  $n$  агентів, які обмінюються певною інформацією, на основі якої приймають подальші рішення (рисунок 4.4).

Вводиться допущення, що  $i$ -й БПЛА функціонує без збоїв і не може бути джерелом негативної інформації.

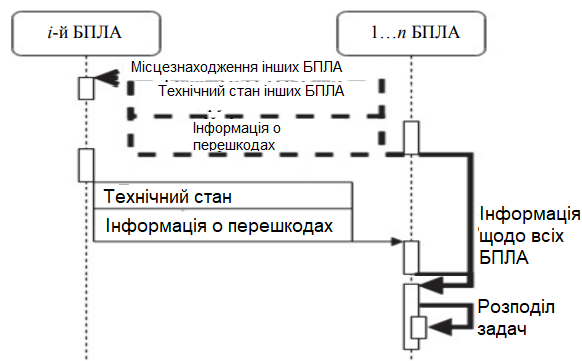


Рисунок 4.4 – Уразливі інформаційні повідомлення та процеси в моделі зовнішнього інформаційного взаємодії елементів в групі безпілотних літальних апаратів

Інформація, що отримується і-м БПЛА від інших агентів, а також інформація, що передається між цими агентами, вразлива для ДІВ. В даному випадку перша інформація передається безпосередньо від агентів, в той час як друга передається і обробляється всіма агентами групи, після чого результат передається всім агентам групи [26].

### 4.3 Структура загальної кінетичної моделі рою

Щоб розглядати модель в якості кінетичної, об'єкти вважатимуться досить маленькими, щоб їх можна було вважати матеріальними точками, а ефект їх обертання відносно центру мас вважається неважливим для їх загального руху. Використовується нерухома декартова система координат. Одиничними векторами для такої системи є взаємно ортогональна тріада ( $e_1$ ,  $e_2$ ,  $e_3$ ). Позначимо положення точки (рою) у просторі вектором  $r$ . У фіксованих декартових координатах маємо:

$$r = r_1 e_1 + r_2 e_2 + r_3 e_3 \quad (4.1)$$

для швидкості,

$$v = \dot{r} = \dot{r}_1 e_1 + \dot{r}_2 e_2 + \dot{r}_3 e_3 \quad (4.2)$$

для прискорення,

$$\mathbf{a} = \ddot{\mathbf{r}} = \ddot{r}_1 \mathbf{e}_1 + \ddot{r}_2 \mathbf{e}_2 + \ddot{r}_3 \mathbf{e}_3. \quad (4.3)$$

#### 4.3.1 Проектування моделі рою

Ми ставимося до членів рою як до точкових мас, тобто ми ігноруємо їх розміри. Для у кожного члена рою (загалом  $N$ ), рівняння руху буде

$$m_i \dot{\mathbf{v}}_i = m_i \ddot{\mathbf{r}}_i = \boldsymbol{\psi}_i^{mt}(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N), \quad (4.4)$$

де  $\boldsymbol{\psi}_i^{mt}$  представляє сили взаємодії членів рою та цілей, які підлягають відображенню, та інших членів рою.

Алгоритм відображення області виконуються у кілька кроків:

1.Ініціалізація місця розташування цілей, які потрібно відобразити:

$$\mathbf{T}_i = (T_x, T_y, T_z)_i \text{ для } i = 1, 2, \dots, N_T \quad (4.5)$$

де  $N_T$  - кількість цілей.

2. Ініціалізація місця розташування членів рою(БПЛА):

$$\mathbf{r}_i = (r_x, r_y, r_z)_i \text{ для } i = 1, 2, \dots, N_T \quad (4.6)$$

3.Для кожного члена рою  $i$  визначимо відстань до кожної цілі  $j$ :

$$\|\mathbf{r}_i - \mathbf{T}_j\| = [(\mathbf{r}_{ix} - \mathbf{T}_{jx})^2 + (\mathbf{r}_{iy} - \mathbf{T}_{jy})^2 + (\mathbf{r}_{iz} - \mathbf{T}_{jz})^2]^{1/2} \quad (4.7)$$

$i$  напрямок до кожної цілі

$$n_{ij} = \frac{\mathbf{T}_j - \mathbf{r}_i}{\|\mathbf{r}_i - \mathbf{T}_j\|} \quad (10.1)$$

4.Для кожного члена рою визначимо силу взаємодії як функцію відстаней

$$\boldsymbol{\psi}_i^{mt} = F(\|\mathbf{r}_i - \mathbf{T}_j\|, n_{ij}). \quad (4.8)$$

Обчислимо зважений напрямок до кожної цілі

$$\sum_{j=1}^{N_T} w_j n_{ij} j e^{-a\|\mathbf{r}_i - \mathbf{T}_j\|} \quad (4.9)$$

де  $w_j$  - вага, що відображає важливість цієї цілі, а - параметр занепаду, який нормалізується, щоб дати загальний напрямок, у якому рухатися

$$n_i^* = \frac{N_i}{\|N_i\|} \quad (4.10)$$

Потім сили будуються шляхом множення наявної тяги на  $F_i$  на загальний нормований напрямок:

$$\psi_i^{mt} = F_i n_i^*. \quad (4.11)$$

5.Інтегрування рівняння руху:

$$m_i \dot{v}_i = \psi_i^{mt} \quad (4.12)$$

Отримаємо

$$v_i(t + \Delta t) = v_i(t) + \frac{\Delta t}{m_i} \psi_i^{mt}(t) \quad (4.13)$$

та

$$r_i(t + \Delta t) = r_i(t) + \Delta t v_i(t) \quad (4.14)$$

Відзначимо, що якщо

$$\|v_i(t + \Delta t)\| > v_{\max} \quad (4.15)$$

то ми визначаємо  $v_i^{old}(t + \Delta t)$  як  $v_i(t + \Delta t)$  та швидкість буде перераховано:

$$v_i^{new}(t + \Delta t) = v_i(t + \Delta t) \frac{v_{\max}}{v_i^{old}(t + \Delta t)}, \quad (4.16)$$

де  $v_i(t + \Delta t) = v_i^{new}(t + \Delta t)$

6. Визначимо, чи були відображені якісь цілі, перевіривши відстань між роєм члени та цілі:

$$\|r_i - T_j\| \leq \alpha, \text{ де } \alpha - \text{точність вимірування.}$$

Для будь-якого  $T_j$ , якщо будь-який член рою задовольнив цей критерій, виймаємо  $T_j$  із системи наступного кроку, щоб жоден член рою не витрачав ресурси, намагаючись відобразити  $T_j$ .

7. Потім процес повторюється для наступного етапу.

Ми прийнемо наступні параметри:

					ІА-62.080БАК.002	Аркуш
Зм	Лист	№ докум.	Підп.	Дата		57

- ваги  $w_i = 1$ ,
- параметр занепаду  $a = 0,01$ ,
- відношення тяги до маси  $F_i / m_i = 105 \text{ Н / кг}$ ,
- 100 членів рою,
- 100 цілей,
- $T = 50 \text{ с}$ ,
- $\Delta t = 0,001 \text{ с}$ ,
- початкова швидкість рою  $v_i(t = 0) = 0 \text{ м / с}$ ,
- початкова область рою (10 м, 10 м, 10 м),
- ділянка відображення (500 м, 500 м, 10 м)
- максимальна швидкість члена рою  $v_{\max} = 100 \text{ м / с}$ .

#### 4.3.2 Узагальнення структура взаємодіючого рою

Для кожного члена рою (загалом  $N_s$ ) рівняння руху є

$$m_i \ddot{r}_i = \psi_i(r_1, r_2, r_3 \dots r_{N_s}), \quad [26], (4.17)$$

де  $\psi_i$  представляє силу взаємодії між членами рою, цілями, перешкодою, та іншими членами рою. Розглянемо наступне розкладання сил взаємодії,

$$\psi_i = \psi_i^{mm} + \psi_i^{mt} + \psi_i^{mo}, \quad (4.18)$$

де між членами рою(член-член) ми маємо

$$\psi_i^{mm} = \sum_{j \neq i} N_s \left[ (\alpha_1^{mm} \|r_i - r_j\|^{\beta_1^{mm}} - \alpha_2^{mm} \|r_i - r_j\|^{\beta_2^{mm}}) \frac{r_j - r_i}{\|r_i - r_j\|} \right], \quad (4.19)$$

де  $\alpha_1^{mm} \|r_i - r_j\|^{\beta_1^{mm}}$  - це притягання,  $\alpha_2^{mm} \|r_i - r_j\|^{\beta_2^{mm}}$  - відштовхування,  $\frac{r_j - r_i}{\|r_i - r_j\|}$  - одиничний вектор. Тут  $\| \cdot \|$  являє собою евклідову норму, а нормований напрямок визначається значенням різниці в векторів положення центрів частинок:



$$n_{ij} \stackrel{\text{def}}{=} \frac{r_j - r_i}{\|r_i - r_j\|} \quad (4.20)$$

Між членами рою і ціллю ми маємо (член-ціль)

$$\psi_i^{mt} = (\alpha_1^{mt} \|r_i - T\|^{\beta^{mt}}) \frac{T - r_i}{\|r_i - T\|} \quad (4.21)$$

та для відштовхування між членами рою та перешкодами (член - перешкода), ми маємо

$$\psi^{mo} = - \sum_{j=1}^q \left[ (\alpha^{mo} \|r_i - O_j\|^{\beta^{mo}}) \frac{O_j - r_i}{\|r_i - O_j\|} \right] \quad (4.22)$$

де  $q$  - кількість перешкод і де всі параметри представлені в кожному рівнянні у  $\alpha$  і  $\beta$  невід'ємними. Зауважимо, що обрана тут конкретна конфігурація сил між елементами лише одна з багатьох можливих для моделювання взаємодії.

Зауважимо, що навколишнє середовище (наприклад, рідина, така як повітря) є джерелом гасіння(демпфування) для системи, бо являє собою супротив. Найпростіша модель має форму (для члена рою  $i$ ):

$$\psi_i^{env} = -c^{env}(v_i - v^{env}), \quad (4.23)$$

де  $v_i$  - швидкість  $i$ -го члена, а  $v^{env}$  - локальна швидкість навколишнього середовища.

Підсумовуючи, ми маємо наступні сили, що діють на кожного члена рою:

$$\psi = \psi^{mt} + \psi^{mt} + \psi^{mo} + \psi^{env} \quad (4.24)$$

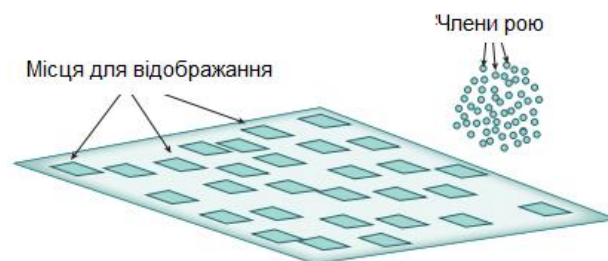


Рисунок 4.5 – Типова модель відображення області

#### 4.3.3 Переслідування рухомої цілі

Розглянемо нормалізовану функціональну функцію (нормалізовану за загальним часом моделювання та початковою дистанцією поділу), що представляє як час, необхідний членам рою для досягнення цілі, так і відстань членів рою до цілі:

$$\Pi = \frac{\int_0^T \sum_{i=1}^{N_s} \|r_i - T\| dt}{T \sum_{i=1}^{N_s} \|r_i(t=0) - T\| dt} \quad (4.25)$$

де загальний час моделювання  $T = 30$  с, а  $T$  - положення цілі. Ці компоненти початкових векторів положення неперервних членів рою, кожному присвоєні масою 10 кг, були задані випадкові значення  $-1 \leq r_{ix}, r_{iy}, r_{iz} \leq 1$ . Положення цільової функція була задана наступною функцією:

$$T_x = x_0 + a_1 \cos(a_2 t) + a_3 t \quad (4.26)$$

$$T_y = y_0 + b_1 \cos(b_2 t) + b_3 t \quad (4.27)$$

$$T_z = z_0 + c_1 \cos(c_2 t) + c_3 t, \quad (4.28)$$

де параметри наведені в таблиці 4.1.

Розташування центру (прямокутного) масиву перешкод було (1,5; 0; 0). 100та-перешкода була встановлена в масиві  $10 \text{ м} \times 10 \text{ м}$  з відстані 0,2 м між центрами перешкод. Для ілюстрації було використано 200 членів рою. Вибрані параметри:  $\alpha_1^{mm} = 1$ ,  $\alpha_2^{mm} = 1$ ,  $\alpha^{mt} = 200$ ,  $\alpha^{mo} = 100$ ,  $\beta_1^{mm} = 2$ ,  $\beta_2^{mm} = 2$ ,  $\alpha^{mt} = 2$ ,  $\alpha^{mo} = 2$ . Навколишнє середовище демпфірування було встановлено на  $c^{env} = 1$ . Запущено моделювання (рис. 4.6 ), з виконанням продуктивності  $\Pi = 0,2712$ .

Таблиця 4.1 – Розподіл параметрів агентів

$(x_0, y_0, z_0)$	$a_1$	$a_2$	$a_3$	$b_1$	$b_2$	$b_3$	$c_1$	$c_2$	$c_3$
(4,0,0)	1	1	0.5	1	1	0.5	1	1	0.5

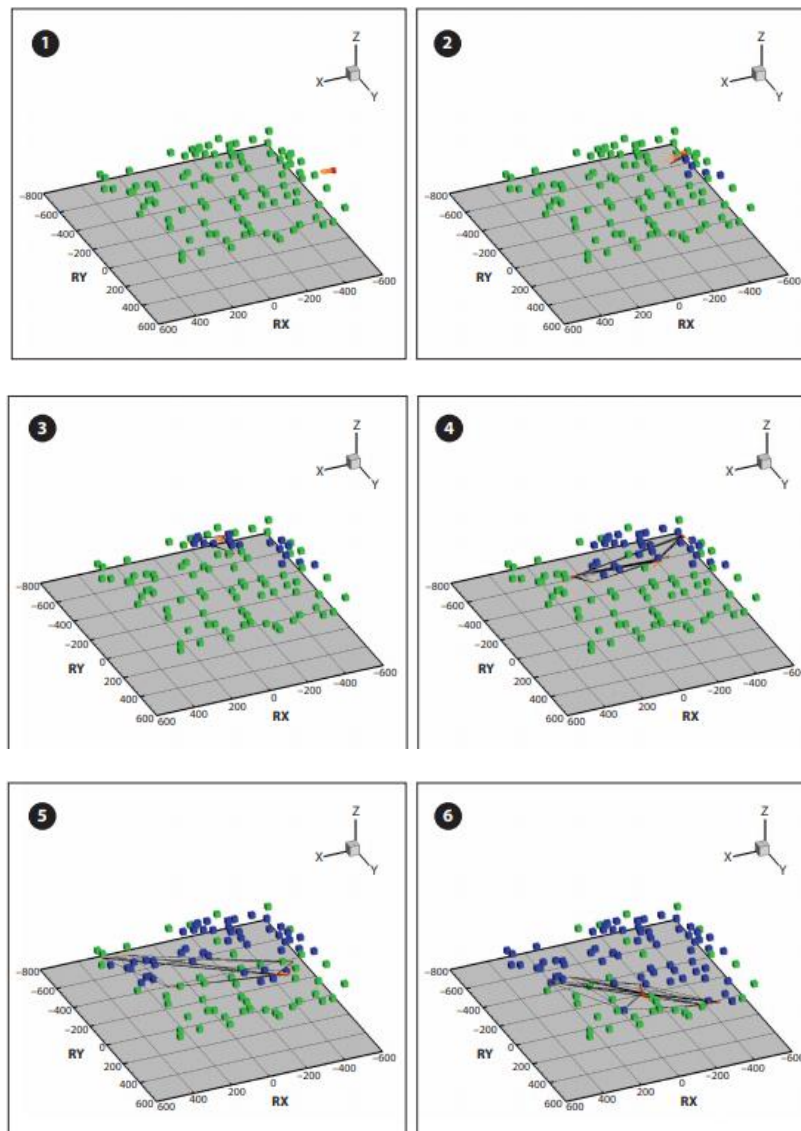
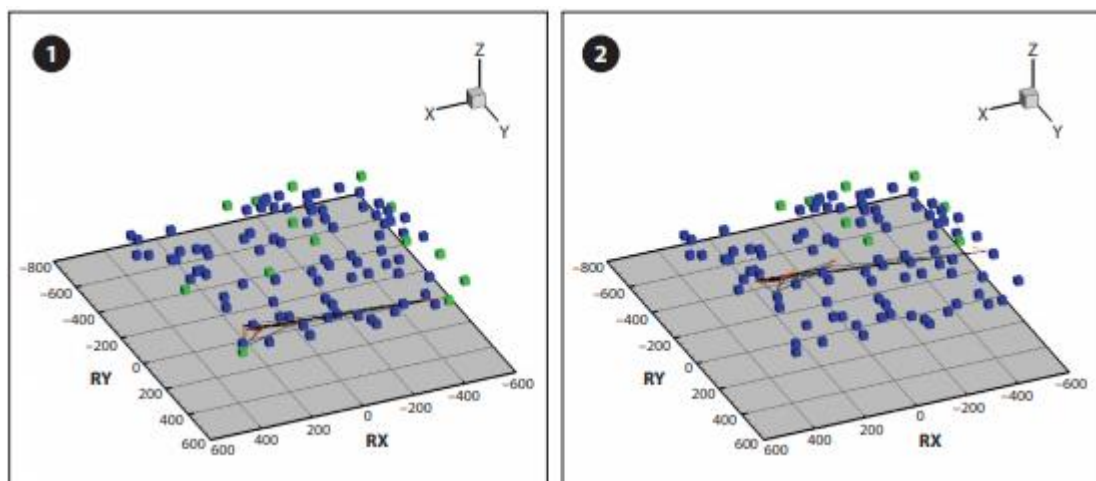


Рисунок 4.6 – Червоні крапки це членів рою, а куби - невідображені (зелені) та відображені (сині) цілі. Векторні стрілки на членах рою представляють швидкість, а лінії кресляться як допоміжний засіб для візуалізації зв'язку між членами рою. Кроки продовжуються рисунку 4.7.



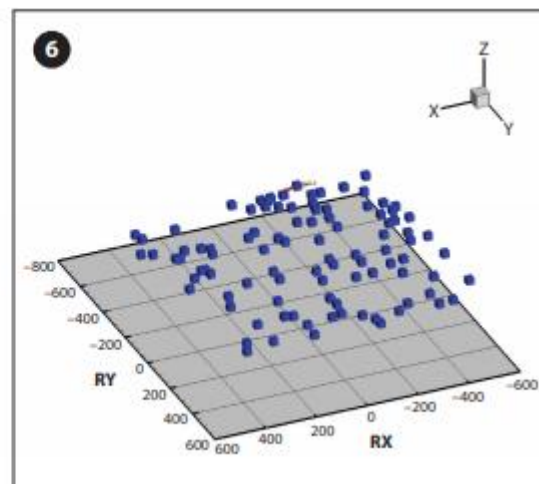
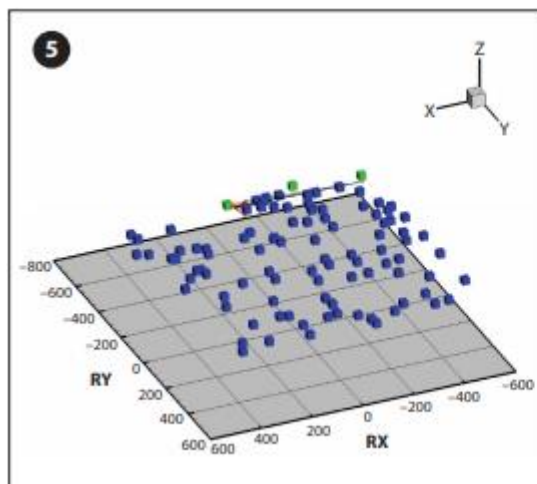
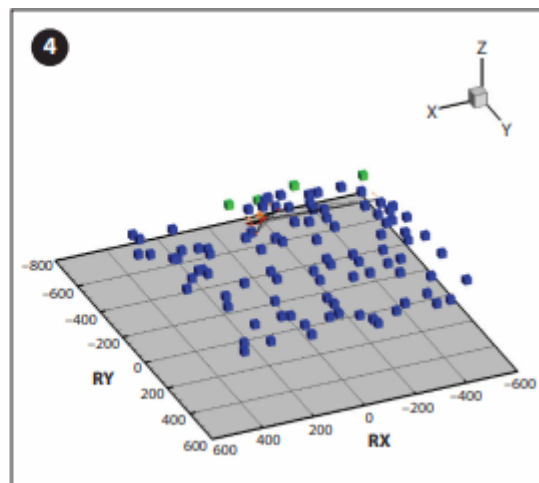
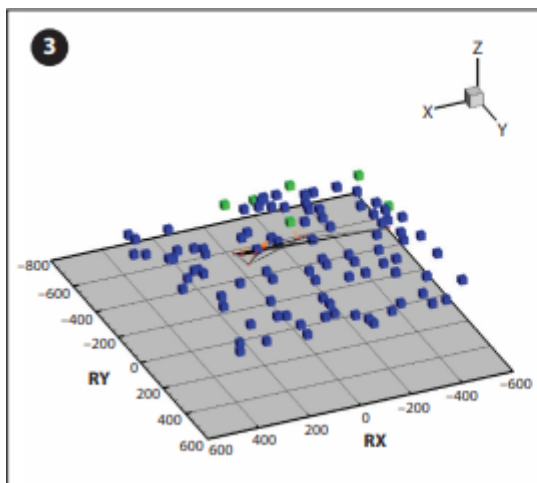
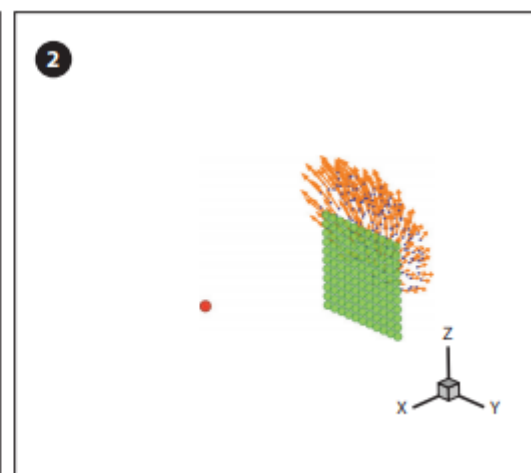
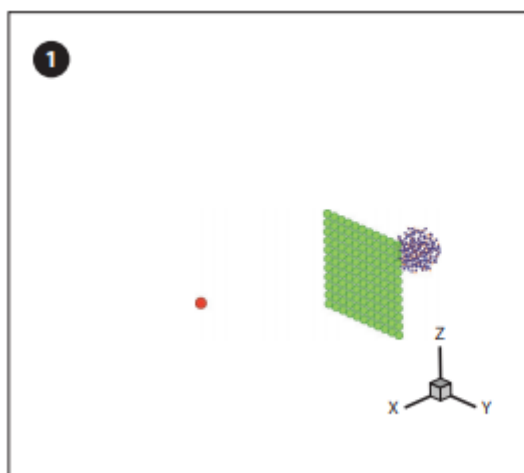


Рисунок 4.7 – Продовження рисунку 4.6



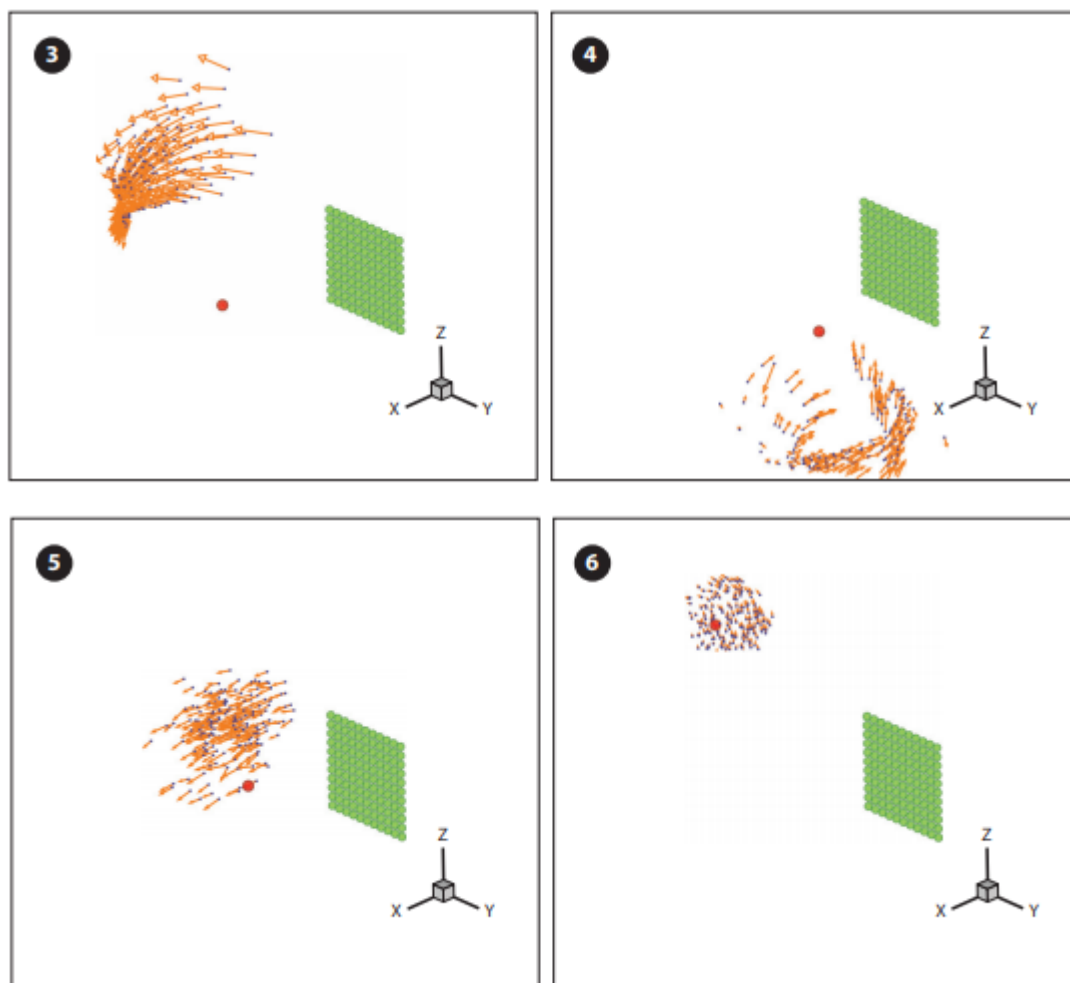


Рисунок 4.8 – Модель пересування рою через паркан перешкод

#### 4.3.4 Багатомісцевий пошук

В якості іншої проблеми моделі розглянемо 400 членів рою та 200 випадково розсіяних цілей. Алгоритм полягає в наступному:

- Кожного члена рою притягують до найближчої цілі;
- якщо місце був відвіданий, то його неактивний рій не приваблює. Як показано на рисунку 4.9, рій має природну схильність до поділу та завоювання ділянки.

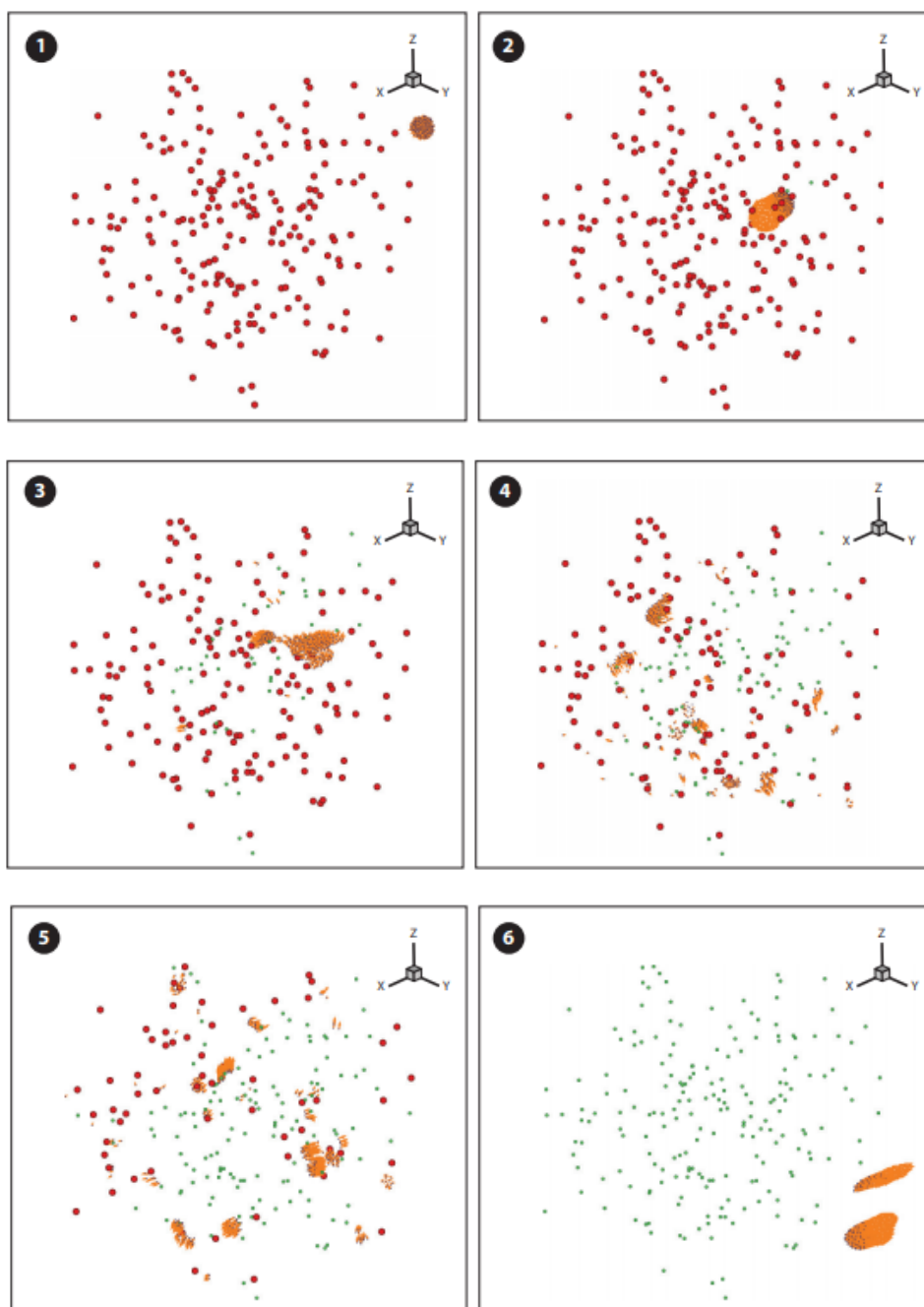


Рисунок 4.9 – Модель поділу рою на групи для займання простору ділянки

#### 4.4 Програмне забезпечення

##### 4.4.1 X-Plane Simulator

Середовище моделювання побудоване навколо симулятора COTS X-Plane. Перевагами використання X-Plane є:

Зм	Лист	№ докум.	Підп.	Дата

**IA-62.080БАК.002**

Аркуш

64

- Сертифікати FAA: X-Plane може надати Федеральне управління авіації (FAA) сертифікаційне моделювання та моделі транспортних засобів. Це дозволяє дослідникам досягти високого рівня впевненості в результатах моделювання.
- Гнучкість операційних систем (ОС): Симулятор X-Plane здатний працювати в ОС Linux, Windows або Macintosh.
- Високорозвинені світові моделі: Цей тренажер отримує постійні вдосконалення вже більше десяти років. Світова модель, що використовується в X-Plane, здатна імітувати, хмарний покрив, дощ, вітер, термічні зміни, мікробухи і туман.
- Перевага успіху: X-Plane був використаний як основа для розробки та тестування алгоритмів управління, які в кінцевому підсумку виявилися життєздатними на фактичному обладнанні в польових експериментах .
- Візуалізація: X-Plane призначений для візуальної співпраці до десяти транспортних засобів. Це означає, що одного агента можна візуалізувати одночасно з одного дисплея, що дозволяє виконувати візуальне налагодження та збирати дані без необхідності додаткового програмного забезпечення.

Основне обмеження X-Plane, як і у більшості подібних програм, - неможливість запускати кілька примірників себе на одному агенті. Це обмеження означає що програма просто не в змозі запустити два однакові процеси одночасно.

Ці машини з'єднані між собою через комутатор, який дозволяє їм передавати дані про стан і керувати будь-яким або всіма машинами в мережі. Така конструкція спричиняє можливе вузьке місце, коли передача даних затоплює мережу, перешкоджаючи адекватній передачі даних.

Стандартним методом зв'язку X-Plane для спілкування із зовнішніми процесами та машинами є протокол User Datagram Protocol (UDP). UDP забезпечує мінімально накладний метод зв'язку для передачі даних між

вузлами. На відміну від протоколу управління передачею (TCP), UDP - це не гарантований протокол і не дає гарантій, що пакети даних прийдуть в порядку або взагалі. UDP призначений для мінімізації використання пропускну здатності, але представляє можливу проблему внаслідок пошкоджених даних.

Вузьке вузьке місце мережі - це врешті-решт питання розуміння апаратних обмежень порівняно з розміром та швидкістю повідомлень, що проходять через мережу. Простіше кажучи, у міру збільшення кількості модельованих транспортних засобів збільшується доступна пропускна здатність у мережі. Використовуючи мережевий комутатор, це середовище моделювання дозволяє уникнути зіткнення пакетів та може скористатися повною медіаною засобів зв'язку з дуплексним обладнанням.

Таким чином, масштабованість мережі обмежується максимальним припливом або відтоком з будь-якого одного вузла мережі. У випадку централізованого управління машина управління буде очевидним вузьким місцем для зв'язку, оскільки всі вузли моделювання будуть безпосередньо спілкуватися з цією машиною. Оскільки і дані про стан, і пакети керування однакові за розміром, число транспортних засобів, яке можна імітувати без перевищення пропускну здатності мережі, можна обчислити, використовуючи:

$$N_{sim} = \frac{R_{net}}{(P_{head} + N_{msg} * D_{size}) * F}, \quad (4.29)$$

де  $R_{net}$  - швидкість комутатора,  $F$  - частота пакетів даних, а розмір пакету UDP складається із заголовка пакета,  $P_{head}$ , розміру окремо вибраного елемента даних стану,  $D_{size}$  та кількості елементів даних стану вибрано,  $N_{msg}$ .

Треба зауважити, що цей розрахунок передбачає оптимальну функціональність мережі та кластеру. Для обладнання ці значення:  $R_{net} = 1000$  Мб,  $P_{head} = 40$  байт,  $N_{msg} = 7$ ,  $D_{size} = 288$  байт і  $F = 50$ . Використовуючи ці дані, максимальна кількість транспортних засобів, що можуть бути імітовані



без перевищення обмежень пропускну́ї здатності, становить 9727 транспортних засобів.

У реальності, контрольний час обчислення та незначні затримки мережевого трафіку зменшить цю кількість. Слід зазначити, що комунікаційні пакети X-Plane, а також швидкість передачі даних вибираються користувачем і можуть змінюватись від даних про положення та швидкості.

Також швидкість передачі даних не може перевищувати швидкість відтворення симулятора. Таким чином, якщо тренажер відтворює лише 25 Гц, то дані про стан будуть обмежені 25 Герцами. Швидкість візуалізації може бути покращена, обмеживши параметри опромінення в симуляторі, зменшивши видимість і видаливши погодні елементи, наприклад хмарний покрив.

#### 4.4.2 OMNeT ++

OMNeT ++ це розширювана, модульна бібліотека та структура C ++ на основі компонентів, в основному для побудови мережевих симуляторів. "Мережа" мається на увазі в більш широкому сенсі, що включає дротові та бездротові мережі зв'язку, мікросхеми, мережі з чергою тощо.

Функціональні можливості, такі як підтримка сенсорних мереж, бездротові спеціальні мережі, Інтернет-протоколи, моделювання продуктивності, фотонні мережі тощо, забезпечуються модельними рамками, розробленими як незалежні проекти. OMNeT ++ пропонує IDE на основі Eclipse, графічне середовище виконання та безліч інших інструментів.

Існують розширення для моделювання в режимі реального часу, мережевої емуляції, інтеграції баз даних, інтеграції SystemC та декількох інших функцій. OMNeT ++ поширюється за ліцензією на академічну публічність.

OMNeT ++ забезпечує компонентну архітектуру моделей. Компоненти (модулі) програмуються на C ++, потім збираються у більші компоненти та моделі за допомогою мови високого рівня (NED).



Рисунок 4.10 – Вікно програми OMNeT ++

Багаторазовість використання моделей виходить безкоштовною. OMNeT ++ має широкую підтримку графічного інтерфейсу, і завдяки модульній архітектурі ядро моделювання (і моделі) може легко вбудовуватися у інші програми.

Основними компонентами OMNeT ++ є:

- Бібліотека ядра імітації (C ++);
- Мова опису топології NED;
- Моделювання IDE на базі платформи Eclipse;
- Інтерактивний графічний інтерфейс моделювання (QtEnv);
- Інтерфейс командного рядка для виконання моделювання (Cmdenv);
- Утиліти (інструмент створення файлів makefile тощо);

- Документація, зразкове моделювання тощо;
- Моделі.

Ядро моделювання OMNeT++ є стандартним для C++ і працює в основному на всіх платформах, де доступний сучасний компілятор C++. Симулятор IDE вимагає Windows, Linux або macOS.

#### 4.4.3 Регулююче програмне забезпечення

Програмне забезпечення управління середовищем моделювання принципово необмежене. Будь-яке програмне забезпечення, здатне безпосередньо або опосередковано приймати та передавати дані UDP, може розглядатися як життєздатний варіант реалізації алгоритмів управління.

Таким чином, в решті цього розділу детально описана модель Matlab / Simulink, яка використовується для перевірки середовища моделювання. Модель Simulink, що використовується для перевірки середовища моделювання, може бути розбита на три основні групи:

- контролер місії: Цей модуль призначений для управління загальним положенням групи шляхом вибору заданої точки в просторі, з якого буде походити гуртування транспортних засобів. Просування від заданої до встановленої точки визначається шляхом порівняння положення кожного транспортного засобу з його індивідуальним бажаним положенням;
- контролер гуртування: Цей модуль відповідає за обчислення індивідуальних заданих значень для кожного транспортного засобу на основі бажаних параметрів гуртування та заданої точки, що надаються контролером місії. Індивідуальні задані значення включають бажану висоту та напрямок, а також двовимірний вектор, що представляє величину помилки та напрямок руху;
- контролер транспортного засобу: модуль управління транспортним засобом відповідає за стабілізацію та навігацію кожного окремого

транспортного засобу. Таким чином, у X-Plane буде створений чіткий модуль управління автомобілем для кожного модельованого агента. Цей модуль включає алгоритми управління, алгоритми обчислення даних про стан та функцію зв'язку XPlane.

#### 4.4.4 Контролер гуртування

Для того, щоб описати контролер гуртування, припустимо, що БПЛА повинні підтримувати конфігурацію еліптичного кільця на встановленій висоті, визначеній вільно розмірами кільця та центром маси. Використовуючи правильно сформоване поле, членів рою можна залучити до еліптичного кільця.

Таке гуртування може бути описана за допомогою послідовності трьох концентричних еліпсів з центром  $(x_c, y_c)$ . На малюнку 3.10 зображено три еліптичні кільця з центром  $(x_c, y_c)$ . Кінцевою метою є залучення членів рою до центрального еліптичного кільця  $R^*$ , описаного як набір точок  $(x, y) \in \mathbb{R}^2$ , що задовольняють:

$$R^{*2} = (x - x_c)^2 + \gamma (y - y_c)^2 \quad (4.30)$$

де  $(x_c, y_c)$  - центр, а  $\gamma$  - відношення осі.

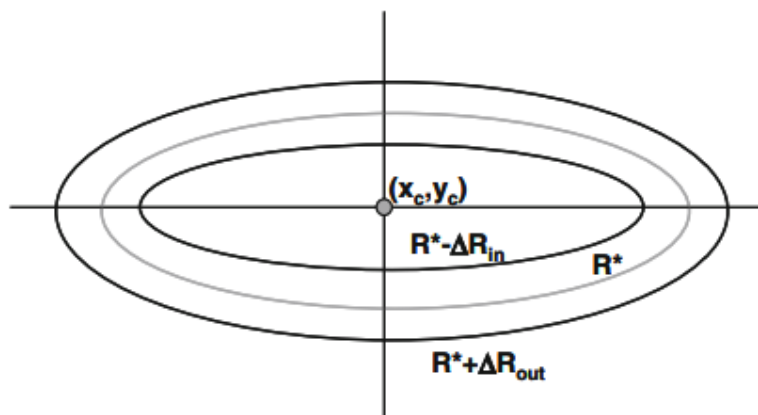


Рисунок 4.10 – Еліптичне притягання об'єднання дронів

Зм	Лист	№ докум.	Підп.	Дата

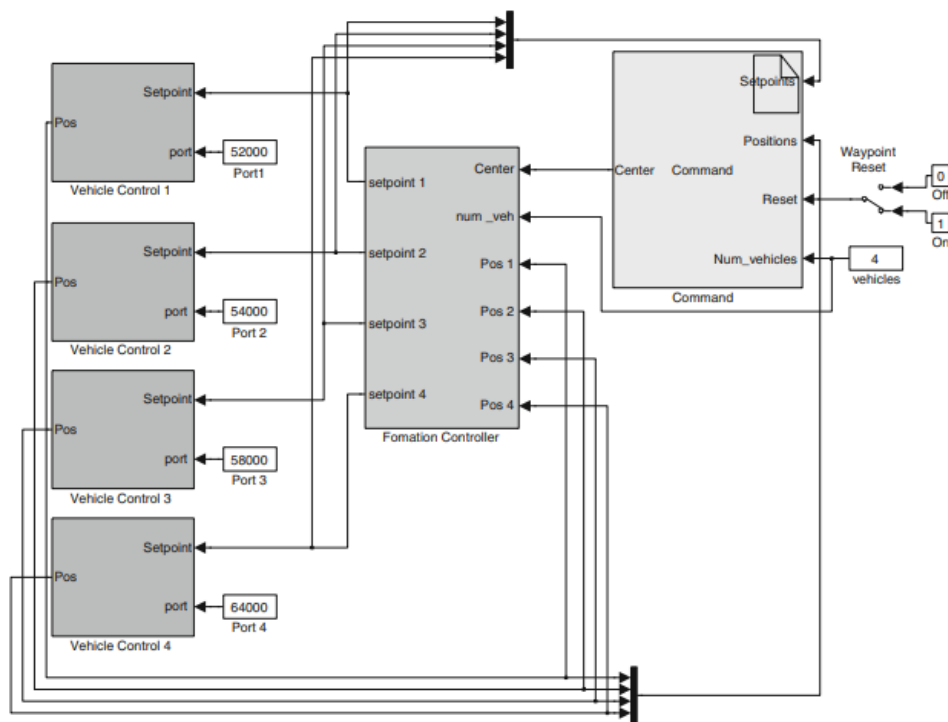


Рисунок 4.11 – Верхній рівень моделі simulink

Загальний контролер гуртування описаний наступним чином:

$$V(x, y, t) = \sum_1^N w_i(x, y, t) V_i(x, y, t), \quad (4.32)$$

де  $V(x, y, t)$  видає швидкість рою в конкретний час і місце. Кожен з векторів  $V_i(x, y, t)$  пов'язаний з різними полями, а  $w_i(x, y, t)$  є вагами загального внеску  $i$ -го вектора. Загалом поле  $V_i(x, y, t)$  - зважена сума  $N$  різних векторів, кожен з яких діє на рій. У цій роботі використовуються три різних векторних поля: одне притягує БПЛА до еліптичної смуги з точок поза еліптичною областю; один відштовхує БПЛА від центру до потрібної смуги; один керує рухом БПЛА всередині смуги. Потенційний контролер на основі поля використовує невелику кількість фізично відповідних ваг  $w_i$  та векторів  $v_i$ , які притягують БПЛА до сусідства еліпса  $R^*$ . Параметри  $R_{in}$  і  $R_{out}$  позначають внутрішню і зовнішню межі  $R$ . Потрібні векторні поля будуть «захоплювати» БПЛА в цих смугах. Як правило, це дуже вузька смуга допустимого простору для БПЛА з керованою шириною  $R_{in} + R_{out}$ , де:

$$R_{in} = R^* - \Delta R_{in} \quad (4.33)$$

$$R_{out} = R^* - \Delta R_{out} \quad (4.34)$$

Векторне поле побудовано з використанням нормованого градієнта. Для кожного  $(x, y)$ , вектор градієнтного поля має вигляд:

$$V_i(x, y) = \begin{cases} V_i(x, y) \frac{1}{L(x, y)} \begin{pmatrix} x - x_c \\ \gamma(y - y_c) \end{pmatrix} & \text{for } (x, y) \neq (x, x_c) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} & \text{для } (x, y) = (x, x_c) \end{cases} \quad (4.35)$$

де

$$L(x, y) = \sqrt{(x - x_c)^2 + \gamma^2(y - y_c)^2} \quad (4.36)$$

Вектор  $\frac{1}{L(x, y)} \begin{pmatrix} x - x_c \\ \gamma(y - y_c) \end{pmatrix}$  - одиничний вектор, який забезпечує напрямок вектора в  $(x, y)$ . Функція  $w(x, y)$  забезпечує величину вектора в цій точці. Для будь-якого  $(x, y)$  цей вектор вказує на центр еліпса.

У визначеному векторному полі частинки, що починаються всередині еліпса  $R^* - \Delta R_{in}$ , мають:

$$R^* = \sqrt{(x - x_c)^2 + \gamma^2(y - y_c)^2} \quad (4.37)$$

рухатись від центру, поки вони не досягнуть району  $R^*$ . БПЛА стартують поза еліпсом  $R^* + R_{out}$  рухаються до центру, поки вони не досягнуть  $R^*$  околиці. Врешті-решт всі БПЛА будуть захоплені у районі:

$$(R^* - R_{in}) \leq r \leq (R^* + R_{out}) \quad (4.38)$$

Уникнення окремих членів рою, включаючи їх розсіювання щодо гуртування контролюється іншою вагою -  $w_i$ , яка обмежує наближення БЛА дозволяти бути один одному. Це визначений користувачем параметр,  $R_{avoid}$ .

#### 4.4.5 Контролер транспортного засобу

Стабілізація та навігація окремих транспортних засобів було зроблено для спрощення процесу перевірки середовища моделювання,

використовуючи контролери, які, як відомо, функціонували належним чином і провели тисячі годин тестування за допомогою симулятора Xplane.

Нечітка логіка була обрана для реалізації контролерів дронів завдяки властивій їй здатності обробляти незначні помилки, шум датчика та суперечливі входи.

Чотири чіткі та нероз'єднані нечіткі контролери використовуються для стабілізації та управління дронами. Вони індивідуально керують рухом, нахилом, рухом та колективом транспортного засобу. Чотири нечіткі контролери використовують постійну нечітку логіку Sugeno і середньозважений метод дефузіфікації, коли вага для всіх правил дорівнює одиниці. Усі правила для контролерів базуються на методі "і" та використовують продукти членства для визначення міцності кожного правила.

Було проведено незначне редагування на двох вхідних змінних, щоб дозволити контролерам відповідати бажаному керуванню кількома транспортними засобами. Більш конкретно, алгоритми управління гуртування були розроблені, щоб забезпечити транспортні засоби потенційними векторами поля, що ведуть їх до гуртування. Нечіткі контролери не були розроблені з урахуванням такого типу вводу. Хоча це і представляло початковий випуск, було визначено, що потенційний вектор поля може розглядатися як такий, що представляє помилку позиції без одиниць. Якщо припустити, що ця невід'ємна помилка положення може бути апроксимована в одиницях довжини, це забезпечує очікуваний вхід в нечіткі контролери.

Це перетворення здійснюється шляхом розділення X і Y компонентів вектора потенційний вектор поля, помножуючи їх на постійну і обертаючи їх на локальну систему координат транспортного засобу,

$$P_E = (-V_x^* \cos(-\psi) + V_y^* \sin(-\psi)) * c \quad (4.39)$$

$$R_E = (-V_x^* \sin(-\psi) + V_y^* \cos(-\psi)) * c \quad (4.40)$$

де  $V_x$  і  $V_y$  - величина потенційного вектора поля на осі  $x$  і  $y$  відповідно,  $\psi$  - кут нахилу Ейлера транспортного засобу, а  $c$  – константа налагоджування регулюючої величини перетворення. Постійна величини,  $c$  це просто спосіб лінійного налагоджування та визначення, скільки помилок має векторна величина. Константа величини була встановлена на сім. Це значення дозволило невеликий діапазон потенційного вектору, який повинен бути перетворений на позиційну помилку з семикратним діапазоном.

## Висновки

1. Таким чином у розділі було приведено ройовий алгоритм як ефективне рішення для забезпечення оптимального контролю за роєм БПЛА.
2. Було проведено аналіз стратегій групового управління.
3. Запропоновано узагальнену модель інформаційної взаємодії.
4. Виконано конструювання рою згідно загальної структури кінетичної моделі.
5. Приведено структуру програмного забезпечення щодо конкретної реалізації.



## 5 МОДЕЛЮВАННЯ УПРАВЛІННЯ РОЄМ ОБ'ЄКТІВ

### 5.1 Симуляція поведінки агентів

Для симуляції було обрано найбільш розповсюдженні сценарії, які б могли посприяти покращенню часових показників у реальних умовах.

#### 5.1.1 Поведінка агентів

Запропоноване завдання складається з спостереження за заданим двовимірним пошуковим простором, який є прямокутною областю, поділеною на комірки фіксованого розміру. Рух агентів обмежений цими клітинами, так що кожному агенту дозволяється переміщатися від центру кожної клітини до центру будь-якої з восьми сусідніх клітин. Це припущення є зручним, оскільки перетворює пошук у дискретну проблему, тим самим, спрощуючи його.

Ми вважаємо, що пошук здійснюється літаючими роботами, мультикоптерами. Агенти здатні літати на постійній висоті та з заданою швидкістю  $v_n$  і оснащені камерою, здатною виявляти цілі в межах кола радіуса  $R_f$ . Ця область називається слід датчика. Команда роботів складається з  $N_a$  з однакових мультикоптерів, спочатку розміщених випадковим чином у позиціях, що не перетинаються в області пошуку. Кожен агент буде налаштований однаково, тобто параметри управління будуть однаковими для кожного.

Розділивши простір пошуку на сітку із загальною площею  $A_s$  ( $m^2$ ), ми дозволяємо роботам переміщатися до будь-якої сусідньої комірки. Для забезпечення повного спостереження за осередками сліду датчика, вони повинні бути вписані в нього. Тому кількість комірок у кожному напрямку становить:

$$n_x = \left\lceil \frac{L_x}{\sqrt{2}R_f} \right\rceil, \quad (5.1)$$

$$n_y = \left\lceil \frac{L_y}{\sqrt{2}R_f} \right\rceil, \quad (5.2)$$

де  $\lceil \cdot \rceil$  - функція стелі, а розміри кожної пошукової комірки:

$$\Delta L_x = \frac{L_x}{n_x} \quad (5.3)$$

$$\Delta L_y = \frac{L_y}{n_y} \quad (5.4)$$

Під час польоту в будь-якому напрямку спостерігається частина сусідніх клітин. Ті клітини, які ми будемо називати відтепер пошуковими клітинками, слід підрозділити на більш тонкі клітинки, щоб врахувати, спостерігали ці ділянки простору чи ні. Ці інші клітини ми будемо називати клітинами дискретизації. Тому рух буде обмежений лише пошуковими клітинами, а не самим пошуковим завданням, оскільки пошукову клітинку можна повністю спостерігати, не відвідуючи її. Ми вважаємо, що розмір осередків дискретизації  $2 \times 2$ м, тобто  $\Delta l_x = \Delta l_y = \Delta l = 2$ м.

У таблиці 5.1 представлені параметри, що визначають сценарії з врахованим діапазоном значень. Треба зауважити, що замість обмеження загальної площі пошуку  $A_s$ , більш розумним є обмеження області пошуку на одного агента,  $A_s / N_a$ . З іншого боку, ми визначаємо співвідношення сторін  $f_A$  пошукового простору як

$$f_A = \frac{L_x}{L_y} \quad (5.5)$$

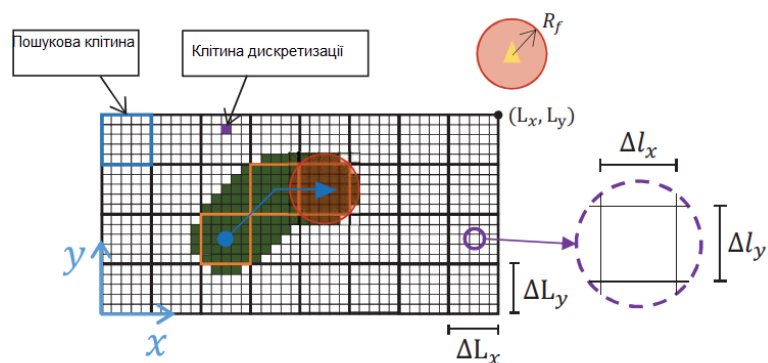


Рисунок 5.1 – Приклад області пошуку.

На рисунку 5.1 виділено клітинку для пошуку та дискретизації. Агент, представлений у вигляді трикутника, перемістився по діагоналі та в горизонтальному напрямку. Виділено три відвідувані пошукові комірки агентом. Помічені осередки дискретизації позначені темно-зеленим кольором. При цьому область сліду обводить пошукові комірки.

Таблиця 5.1 – Параметри, що визначають умови сценарію

Параметр	Опис	Діапазон значень
$A_s / N_a$	Область пошуку на одного агента	$[2, 15] \times 10^3 \text{ м}^2$
$N_a$	Кількість агентів	$[2, 30]$
$v_n$	Номінальна швидкість агентів	$[2, 20] \text{ м/с}$
$R_f$	Радіус сліду датчика	$[5, 20] \text{ м}$
$f_A$	Співвідношення сторін в області пошуку	$[0.25, 1]$

### 5.1.2 Проектування Регулятора

ПД-регулятор - це стандартний інструмент у процесі сучасної промислової автоматизації завдяки їх гнучкості. Він може бути представлений багатьма формами, зокрема, найважливіша

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} dt \quad (5.6)$$

де  $u(t)$  – змінна регулювання, похибка відслідковування  $e(t)$  визначається як

$$e = W_r - W \quad (5.7)$$

де  $W_r$  - очікуваний вихід і  $W$  - дійсний вихід,  $K$ ,  $K_P$ ,  $K_I$  і  $K_D$  - це коефіцієнти посилення контролера, пов'язані з пропорційними (P), інтегральними (I) та похідними (D) діями, відповідно.

ПД-регулятор:

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} dt \quad (5.8)$$

Тоді значення кутів регулювання буде наступним:

$$\theta_{pi}(t) = \arcsin(K_p^x (x_{pi}(t) - x_i(t)) + K_d^x (\dot{x}_{pi}(t) - \dot{x}_i(t))) \quad (5.9)$$

$$\varphi_{pi}(t) = -\arcsin(K_p^x (p_{pi}(t) - y_i(t)) + K_d^x (\dot{y}_{pi}(t) - \dot{y}_i(t))) \quad (5.10)$$

Тоді використаний ПД-регулятор висоти з нелінійною компенсацією виглядатиме наступним чином:

$$u_{cli}(t) = \frac{1}{\cos \varphi'_i \cos \theta'_i} (K_p^z (z_p - z) + K_d^z (\dot{z}_p - \dot{z}) + mg) \quad (5.11)$$

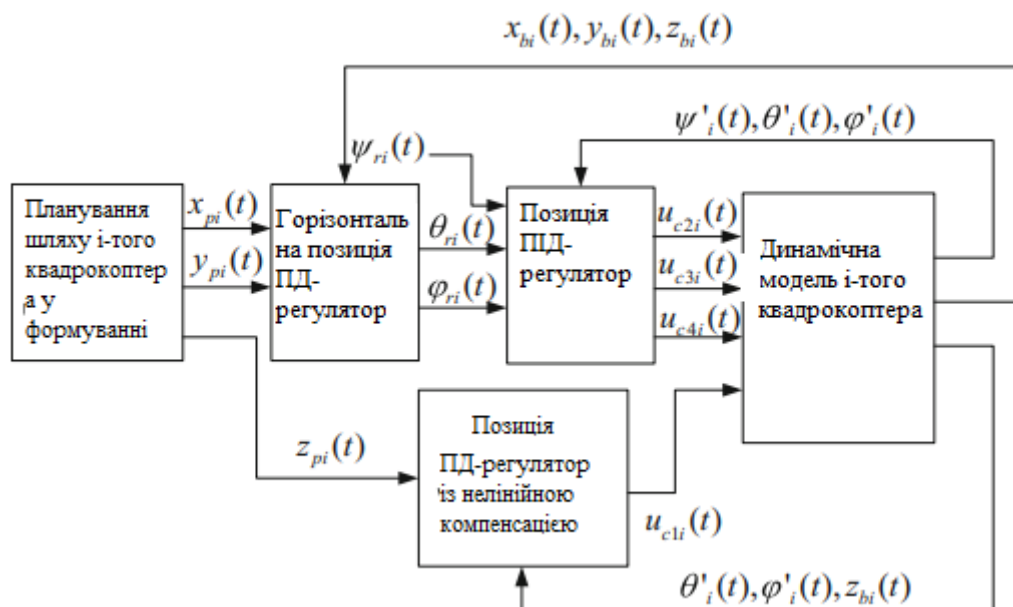


Рисунок 5.2 – Система управління на основі представлених регуляторів

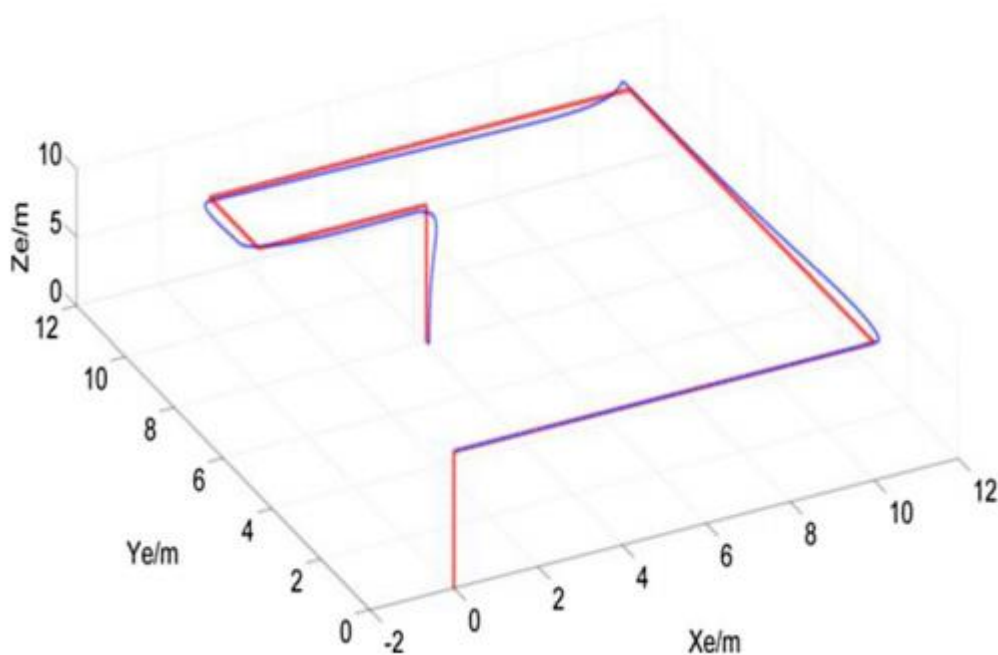


Рисунок 5.3 – Моделювання поведінки слідування із використанням ПД-регулятора

### 5.1.3 Сценарії поведінки

#### 5.1.3.1 Сценарій пошуку

У експерименті ми передбачаємо попередні знання про можливе розташування цілей, які потрібно знайти. Відповідно до цього розподілу генеруються цілі  $N_a$ . Пошук завершується, коли всі цілі були відвідані. Розподіл ймовірностей генерується за допомогою методу переміщення середньої точки, зазвичай використовується як метод генератора рельєфу місцевості.

Початковий розподіл генерується випадковим чином із шорсткістю  $r \in [0, 0,5]$  та його похідною  $r_t \in [0, 0,1]$ . Застосовується п'ять ітерацій. На рисунку 5.4 показаний приклад цього типу сценарію.

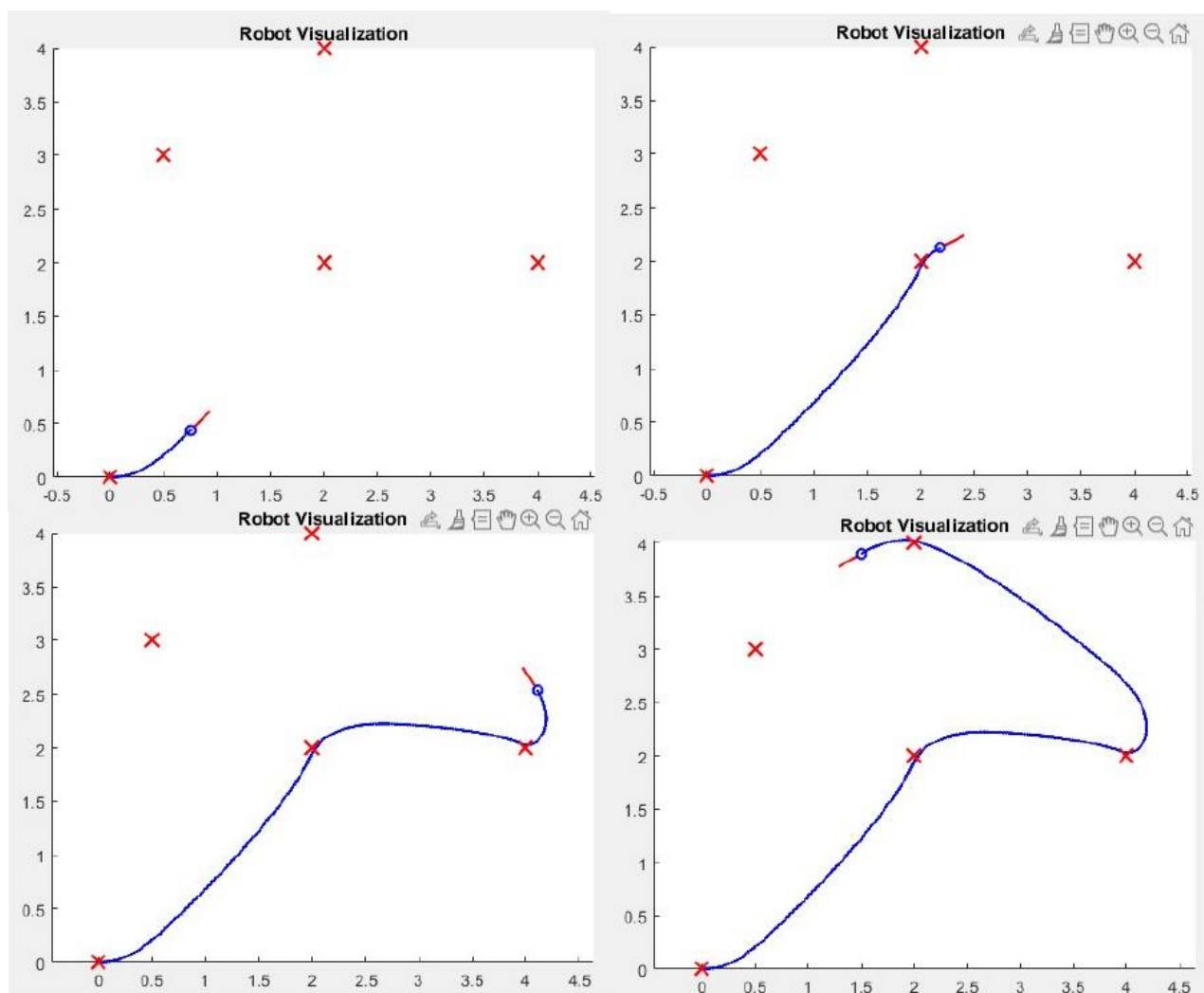


Рисунок 5.4 – Пошук цілей ( $t = 5, t=10, t=15, t=20$ )

### 5.1.3.2 Сценарій з перешкодами

Наступний тип сценарію, що підлягає аналізу, - це сценарій з нелітаючими перешкодами. У цьому випадку БПЛА завдяки датчикам сканують місцевість поблизу та на визначеній відстані здійснюють маневр для того, щоб минути перешкоду. Пропонується прийняти, що кількість агентів – 3, перешкоди являють собою заздалегідь встановлені статичні об'єкти та виставлені в околиці початкового розташування агентів. Мінімальна дистанція до перешкоди – 20 см.

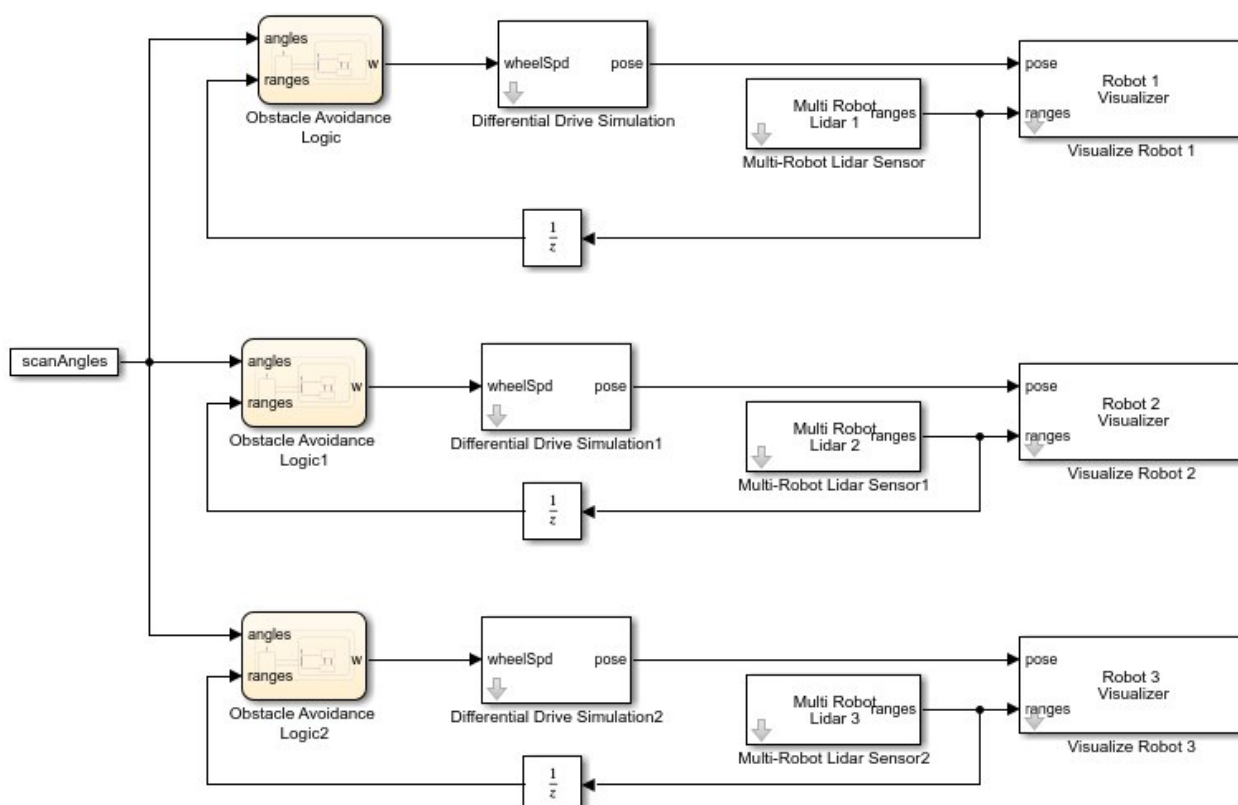


Рисунок 5.5 – Модель у MATLAB для опрацювання поведінки оминання перешкод

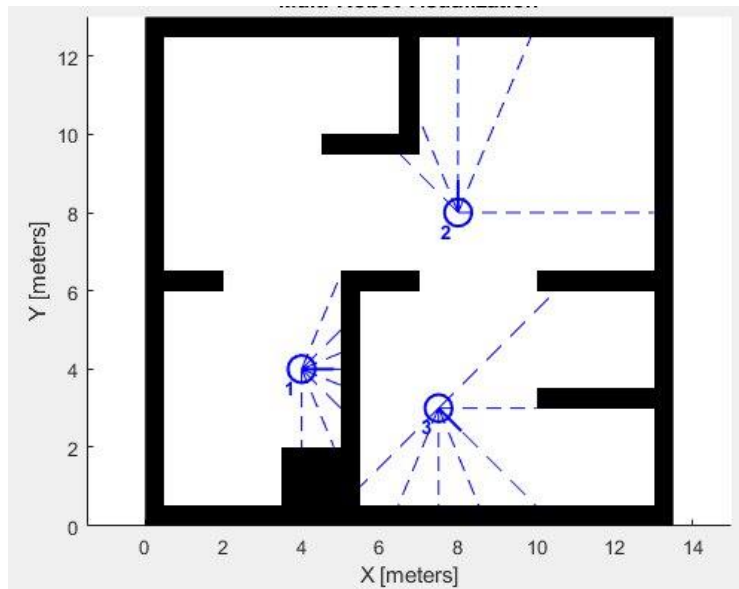


Рисунок 5.6 – Поведінка при опрацюванні перешкод(  $t = 5$ )

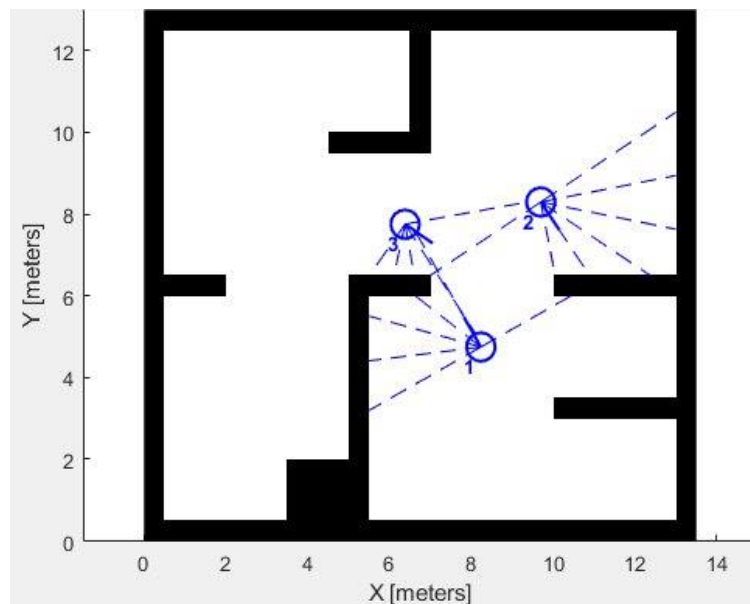


Рисунок 5.7 – Поведінка при опрацюванні перешкод (  $t = 10$ )

### 5.1.3.3 Сценарій із прокладання заздалегідь відомого шляху

Планування шляху використання багатопланових БПЛА використовує початкові параметри, які наведені в таблиці 5.2. У розділі простір без перешкод, результат моделювання гуртування показаний на рисунку 5.11. БПЛА формуються і досягають цільової точки одночасно.



Таблиця 5.2 – Початкові координати при моделюванні

Дрон	(x,y,z)(m)
Лідер(Чорний)	(1500,2000,500)
Дрон1(синій)	(2000,1500,200)
Дрон2 (червоний)	(1500,1500,400)
Дрон3(жовтий)	(1500,2000,500)
Дрон4(зелений)	(1500,4000,500)

У даному сценарії БПЛА виконують завантажену операцію слідування шляху. У блоку «Waypoint» можна змінювати параметри для слідування.

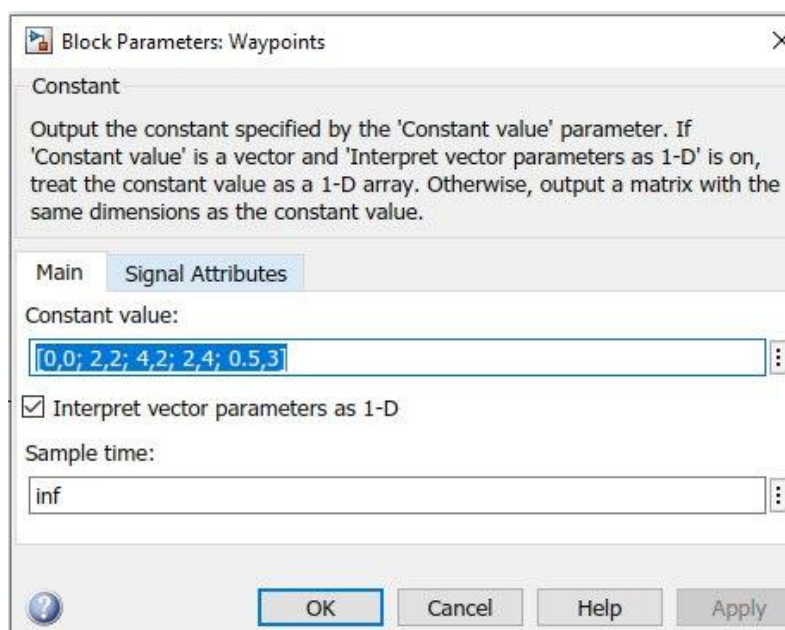


Рисунок 5.8 – Редугування векторів руху

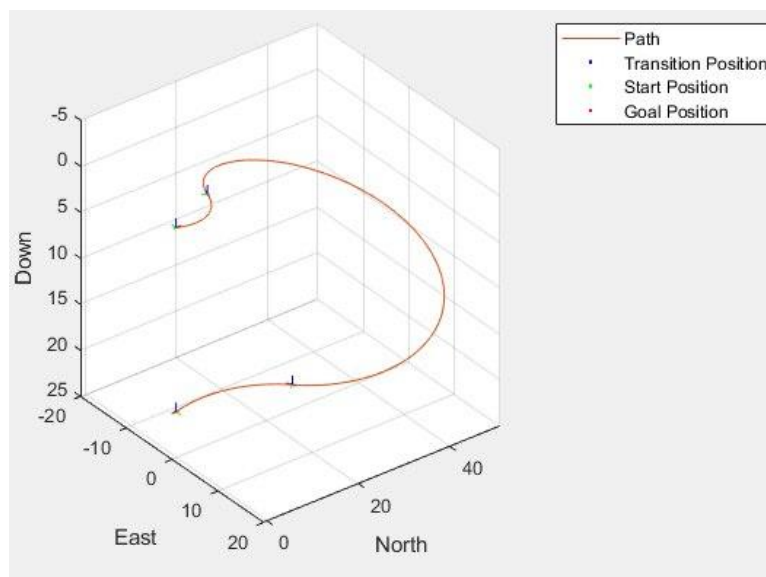


Рисунок 5.9 – Слідування заздалегідь відомого шляху

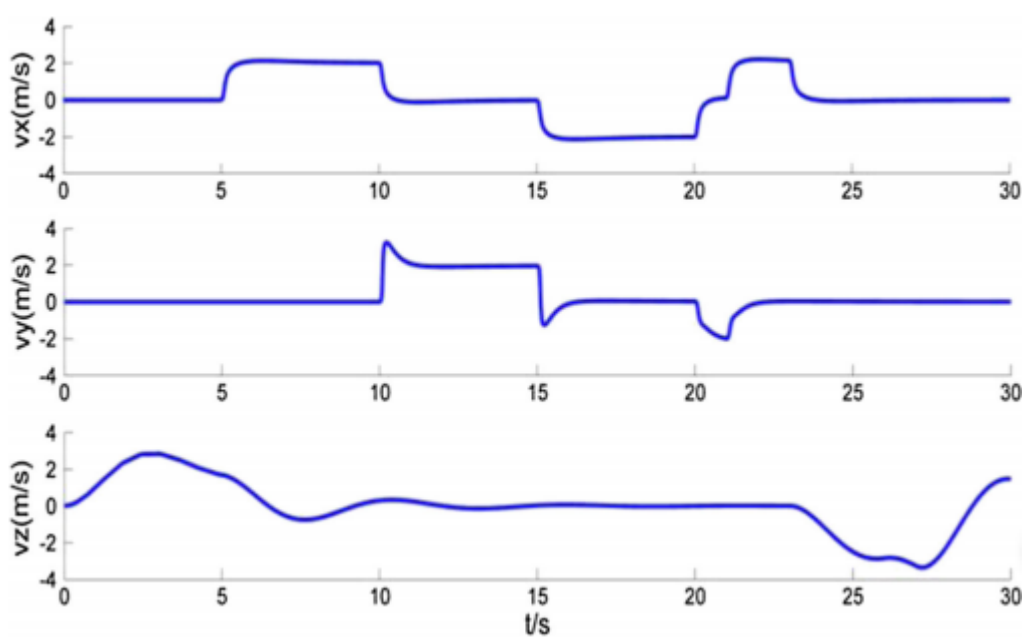


Рисунок 5.10 – Часові діаграми стану польоту

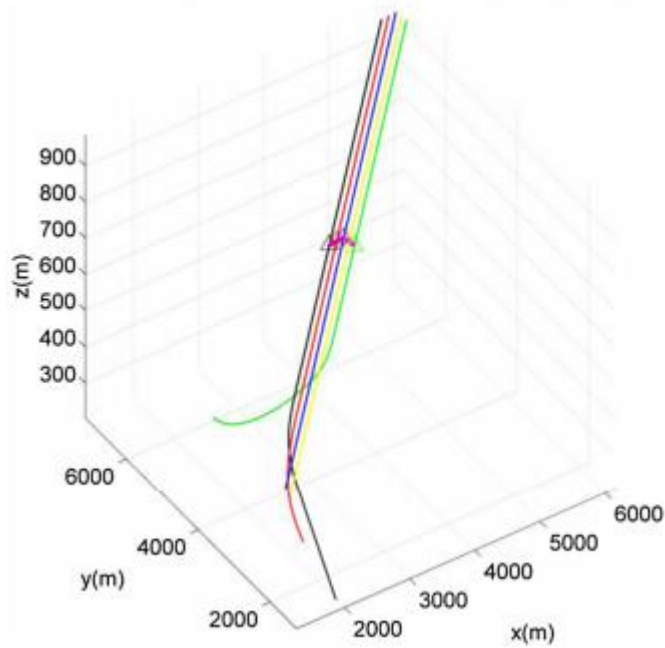


Рисунок 5.11 – Формування єдиного шляху

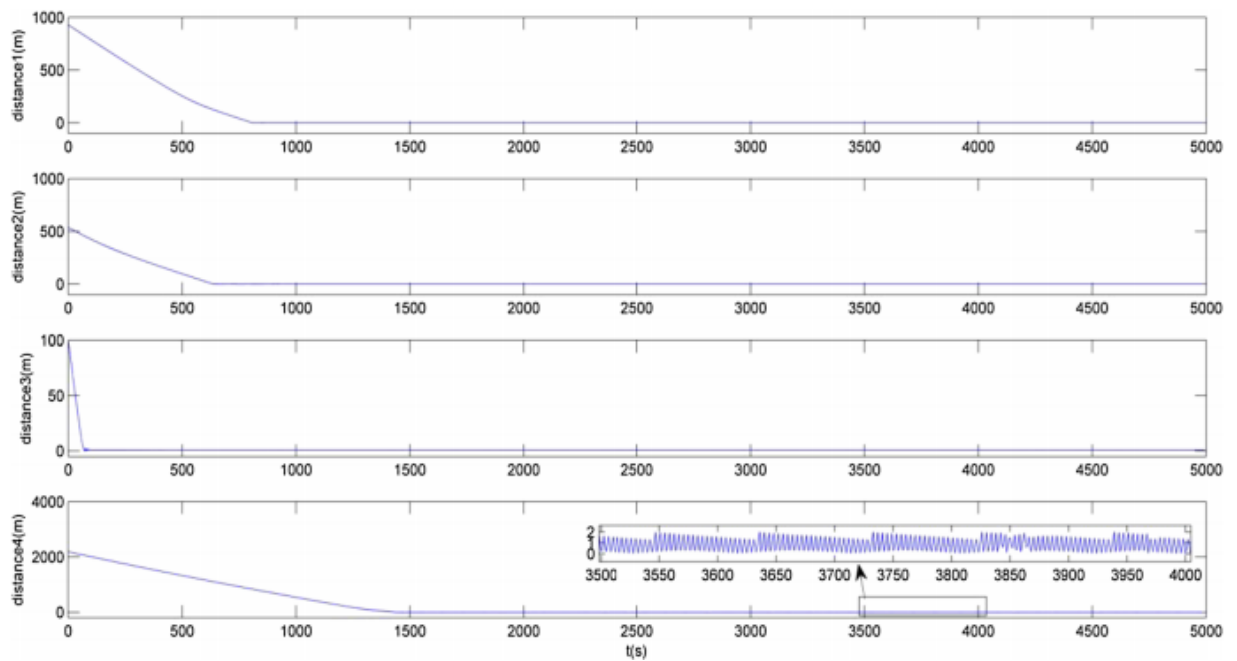


Рисунок 5.12 – Результати похибки гуртування чотирьох БПЛА без перешкод

#### 5.1.3.4 Планування простору симуляції

У складних умовах, складних і фактичний простір планування будується за допомогою інтерполяції MATLAB. До перешкод належать: альпійська місцевість, радарна зона, погана погода тощо. Ці перешкоди є дуже поширені у реальному середовищі поля бою. Конкретна задача полягає в наступному (Рисунок 12.14): Усі перешкоди, включаючи радіолокаційні зони та місцевість в просторі планування розглядаються як зона незальоту, куди не можна допускати траєкторії БПЛА. Горизонтальне обмеження швидкості БПЛА дорівнює  $v_{zm} = 10 \text{ м / с}$ , обмеження вертикальної швидкості БПЛА становить  $v_{zh} = 5 \text{ м / с}$ ; горизонтальне обмеження прискорення БПЛА -  $a_{zh} = 10 \text{ м / с}^2$ , обмеження вертикального прискорення БПЛА становить  $a_{zh} = 10 \text{ м / с}^2$ ; то обмеження висоти польоту БПЛА дорівнює  $z > z$  (поверхня), де  $z$  (поверхня) означає результат інтерполяції планування простору.

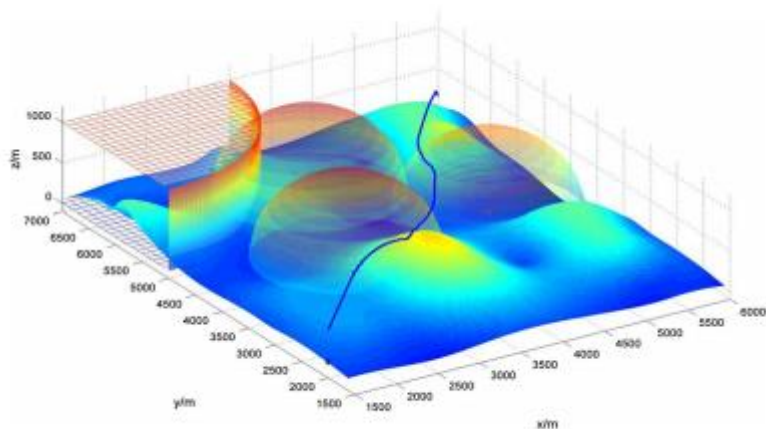


Рисунок 5.13 – Шлях дрона-лідера у місцевості із перешкодами

Результат моделювання показує, що цей метод має добру здатність планування шляху в даному плануванні простору і він долає проблему мертвих зон в процедурі планування шляху. Поєднуючись із простим результатом моделювання, досить перевірити ефективність методу планування єдиного шляху БПЛА.

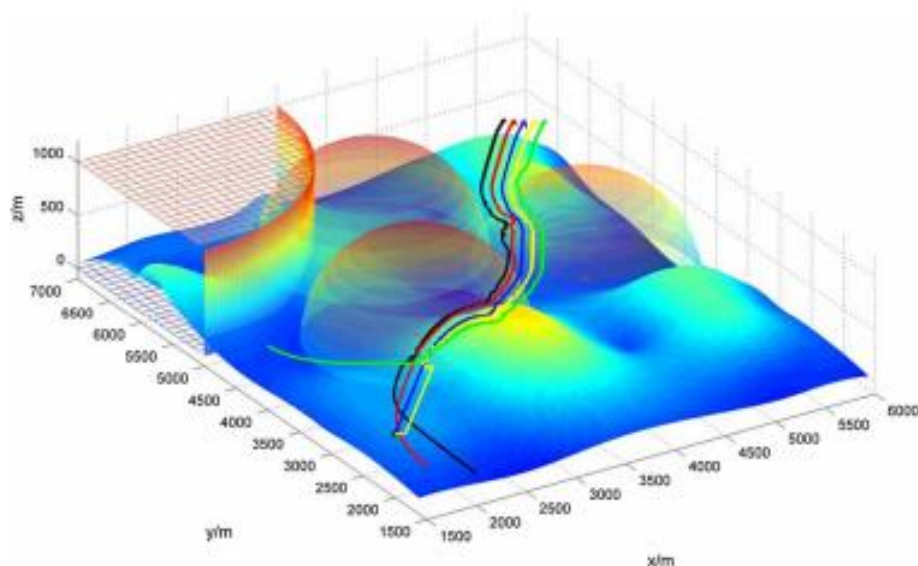


Рисунок 5.14 – Планування шляху гуртування рою БПЛА

На рисунку 5.14 вказується, що БПЛА, встановлені з різних вхідні точок і у процесі польоту гуртуються. БПЛА літають у очікуваному плані та одночасно досягають цільової точки. Аби описати формульовально-польотні показники точно, результати помилок гуртування чотирьох БПЛА показані на Рисунку 5.15.

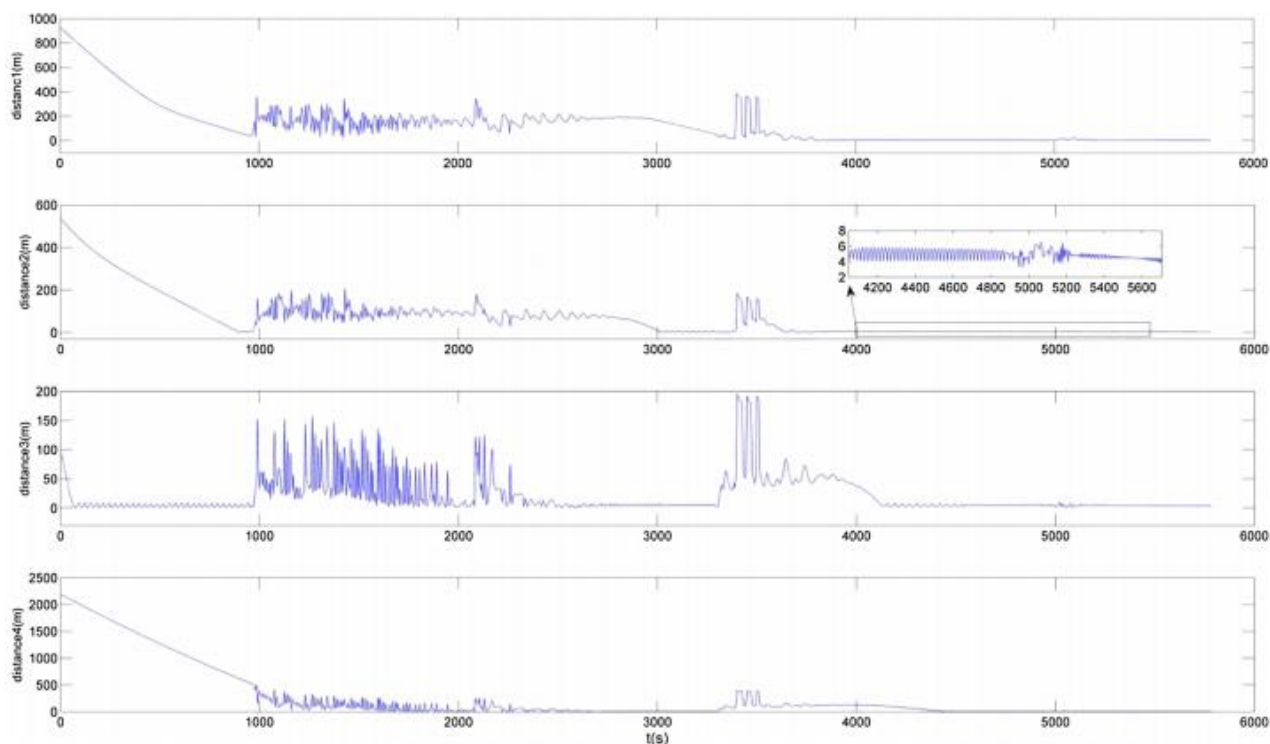


Рисунок 5.15 – Похибка гуртування у комплексному просторі

### 5.1.3.5 Сценарій кластеризації

Для реалізації наступного сценарію нам необхідно пребігти до виконання MATLAB – скрипту. У даному сценарії ми симулюємо випадкову генерацію БПЛА у просторі у кількості 20 штук. Кількість кластерів дорівнює 5.

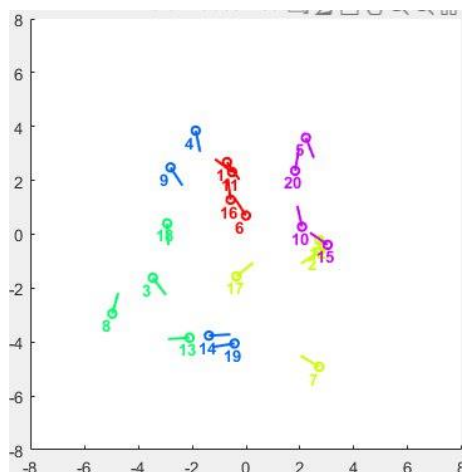


Рисунок 5.16 – 20 БПЛА у випадковій послідовності(  $t = 5$ )

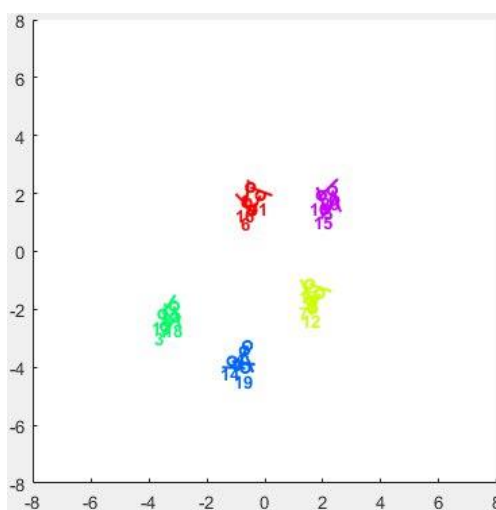


Рисунок 5.17 – 5 кластерів БПЛА (  $t = 25$ )

#### 5.1.3.6 Сценарій польоту в умовах обмежених ресурсів

Вхід: Простір представлено у вигляді повного зваженого графа  $G = (N, E)$ , де  $N$  - сукупність вузлів,  $E$  – множина ребер, що з'єднують вузли. Набір вузлів  $N$  включає в себе а набір базових станцій  $N_b$  і набір точок зображення  $N_p$ . Кожне ребро  $e_{i,j} \in E$  має свою вагу  $d_{i,j}$  - відстань між вузлами  $i$  та  $j$ , які пов'язані цим ребром. У присутності перешкод, ця вага дорівнює довжині найкоротшого шляху між вузлами.

Кожна точка зображення  $p \in N_p$  має дві властивості:

- пріоритет  $pr_p$  позначає важливість точки  $p$ , тобто чим нижче значення пріоритету, тим менш важливим є цей момент;
- різниця у часі  $lvt_p$  між останнім відвідуванням точки  $p$  та початку місії вказує на те, як довго точка залишалася неінспектованою до початку моніторингу. Ми ввели  $lvt_p$  для перепланування сценаріїв, де  $lvt_p$  та ін. параметри оновлюються перед початком переплануванням.  $lvt_p$  дорівнює нулю для всіх точок, якщо жодна з них не була проінспектована раніше.

Якщо лише деякі точки не мають попередніх відвідувань, їх  $lvt_p$  присвоюється максимальним значенням  $lvt_p$  відвідуваних точок.

Кожна базова станція  $b \in N_b$  характеризується числом батарей  $nBat_{b,t}$ , для транспортних засобів типу  $t \in T$ . Загальна кількість батарей типу  $t$  позначається як  $nBat_t$ . Кожен транспортний засіб  $v \in V$  характеризується своїм типом  $t_v \in T$ , початкова залишкова ємність акумулятора  $inRemCap_v$  та початкове розташування  $inLoc_v \in N$ . Початкове місце розташування може бути або як на базовій станції або точкою на зображенні.

Транспортні засоби типу  $t \in T$  мають певну середню швидкість  $sp_t$ , ємність батареї  $batCap_t$ , задана як максимальний час польоту в одиницях часу, сервісний час на різноманітні функції  $servTt$  (наприклад, зробити фотографії в точці зображення) та час зміни батареї  $tChBatt$ .

Вихід: Рішення проблеми - це послідовність маршрутів для кожного транспортного засобу  $v \in V$ , тобто  $R_v = (R_v, 1, \dots, R_{v,nR_v})$ , де

$nR_v$  - кількість маршрутів транспортного засобу  $v$ . Кожен маршрут це послідовність вузлів  $R_{v,y} = (r_{v,y}, 1, \dots, r_{v,y}, k)$ , де  $k$  дорівнює кількості відвідувань транспортним засобом  $v$ ;  $r_{v,y,2}, \dots, r_{v,y,k-1}$  є точками зображення. Вузли  $r_{v,y,1}$  і  $r_{v,y,k}$  - базові станції(charging station(CS)) за винятком першого пункту першого маршруту, який дорівнює початкове розташування транспортного засобу, тобто  $r_{v,1,1} = inLoc_v$ . Можливе рішення повинно відповідати наступним вимогам:

1. час польоту кожного маршруту не перевищує встановленої ємності акумулятора;
2. кожен вузол відвідує не більше одного транспортного засобу за раз;
3. транспортний засіб може змінити свій акумулятор тільки на базові станції з наявними запасними батареями відповідного типу;
4. після закінчення місії всі дрони повинні бути на станціях.



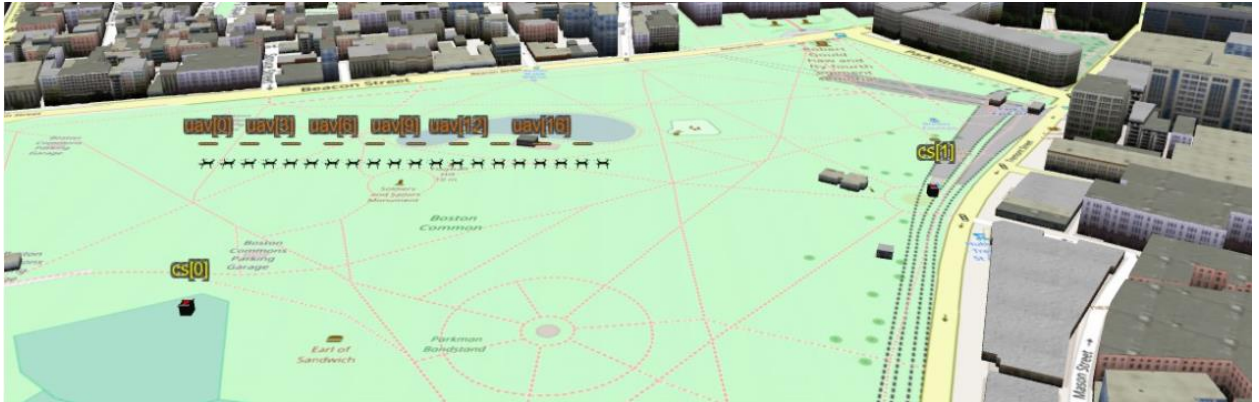


Рисунок 5.18 – Моделювання рою із двома зарядними станціями

На рисунку 5.18 зображено початкові умови при моделюванні рою БПЛА. Кількість агентів – 20, зарядних станцій – 2.

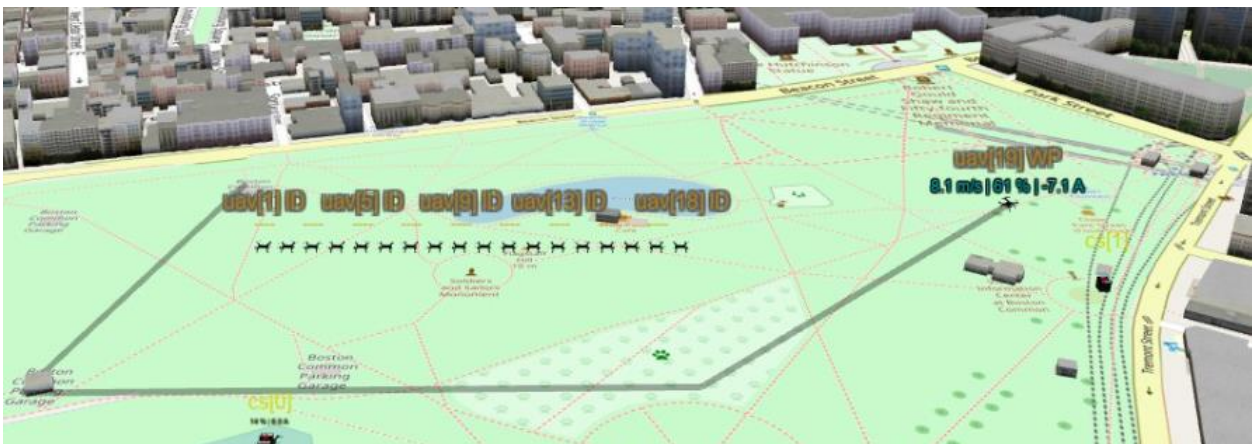


Рисунок 5.19 – Прокладання шляху відповідно розташування зарядних станцій і початкових позицій дронів

На рисунку 5.19 зображено певний крок стану моделювання, при якому робота,- прокладання шляху БПЛА базується на місцезнаходженні зарядних пристроїв(CS). У даному випадку максимальна віддаленість конкретних агентів буде залежати від взаємного розташування зарядних станцій і шлях, прокладений агентами виконується згідно оптимального часу для повернення до найближчої станції.

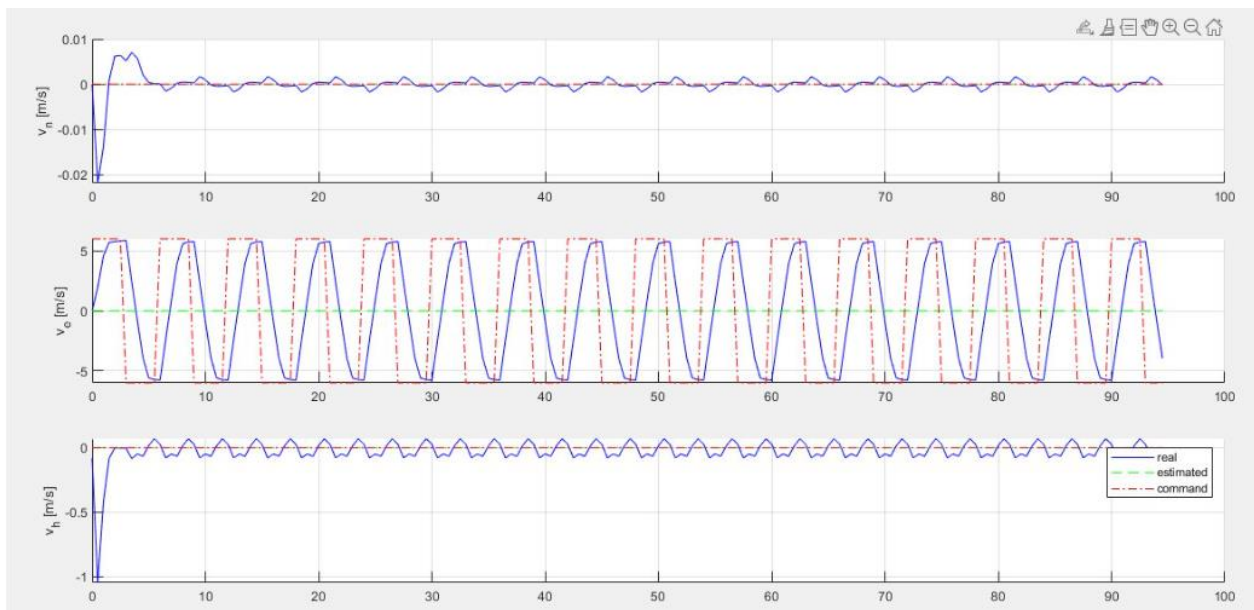


Рисунок 5.20 – Часові характеристики швидкості

Дана часова діаграма ілюструє швидкісну характеристику системи та ступінь її збіжності із бажаним значенням. Згідно неї, система функціонує досить стабільно, але все ж таки є недоліки у вигляді невідповідності бажаним характеристикам, а саме –  $v_h$  – 15,7 %,  $v_o$  – 13,2%,  $v_n$  – 11,4%. Взагалом систему можна вважати стабільною.

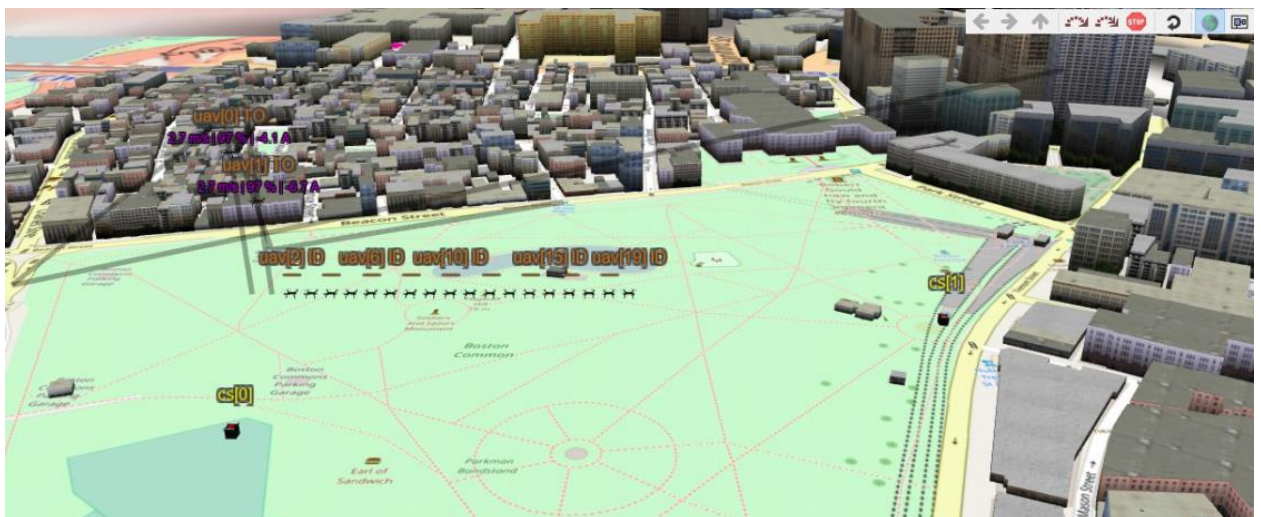


Рисунок 5.21 – Прокладання шляху відповідно розташування зарядних станцій і позиції дронів [0],[1]

У даному випадку проілюстровано як поводитиме себе система в ситуації зміни зарядного пристрою. Зміні піддається переформування членів роя і поведінки, а саме, прокладання певного заданого шляху відповідно до її місцезнаходження.



Рисунок 5.22 – Розрахунок оптимального шляху

## Висновки

1. Було виконано симуляцію найбільш поширених сценаріїв поведінки рою БПЛА та приведено часові діаграми до відповідних сценаріїв, які ілюструють ефективність рішення.

2. У деяких випадках отримано виграш у швидкості у середньому на 14% порівняно із конкурентними рішеннями, за допомогою використання більш сучасних алгоритмів ройового управління. При моделюванні було дотримано вимог щодо розташування агентів таким чином, щоб забезпечити максимальну швидкодію.

## ВИСНОВКИ

Було приведено порівняльний аналіз вже існуючих східних рішень по забезпеченню контролю за БПЛА. Як за ідею роботи було прийнято рішення взяти найкращі нароботки із існуючих рішень та на їх основі відмоделювати рій БПЛА. Було виведено основні принципи чому рій у природі є настільки ефективними. Було розроблено математичну модель (ММ) об'єкта дослідження відповідно до умов, наближених до реальних. Кінцеві результати по моделюванню було представлено у вигляді функціональних схем у MatLab/Simulink. Було проведено аналіз стратегій групового управління та запропоновано узагальнену модель інформаційної взаємодії. Виконано конструювання рою згідно загальної структури кінетичної моделі. Було виконано симуляцію найбільш поширених сценаріїв поведінки рою

					<b>ІА-62.080БАК.002</b>	Аркуш
Зм	Лист	№ докум.	Підп.	Дата		94



БПЛА та приведено часові діаграми до відповідних сценаріїв, які ілюструють ефективність рішення. У деяких випадках отримано виграш у швидкості у середньому на 14% порівняно із конкурентними рішеннями, за допомогою використання більш сучасних алгоритмів ройового управління. При моделюванні було дотримано вимог щодо розташування агентів таким чином, щоб забезпечити максимальну швидкодію.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Román Estebn Hofer. Preliminary design of an unmanned aerial vehicle with application to photovoltaic farm inspection. 2014/2015.
2. Саймон Д. Алгоритми еволюційної оптимізації. - М: ДМК Пресс, 2020. - 940 с.
3. Gould J.L. Honey bee recruitment: the dance-language controversy. Science 189 (4204): 685-693, 1975
4. Hölldobler B., Wilson E. O. The Ants. — Harvard University Press, 1990. — 732 p.
5. Субботін С.О., Олійник А.О., Олійник О.О. Неітеративні, еволюційні та мультіагентні методи синтезу нечіткологічних і нейромережних моделей: Монографія / Під заг. ред. С.О. Субботіна. – Запоріжжя: ЗНТУ, 2009. – 375 с.
6. Román Estebn Hofer. Preliminary design of an unmanned aerial vehicle with application to photovoltaic farm inspection. 2014/2015.

7. Yi Liu, Junyao Gao, Cunqiu Liu, Fangzhou Zhao and Jingchao Zhao Reconfigurable Formation Control of Multi-Agents Using Virtual Linkage Approach Reprinted from: Applied Sciences 2018, 8, 1109
8. Попов В.А., Федутин Д.В. Развитие направления миниатюрных беспилотных летательных аппаратов за рубежом // ФГУП «ГосНИИАС» ([http://uav.ru/articles/mav\\_abroad.pdf](http://uav.ru/articles/mav_abroad.pdf)). 5. Dorigo M., Birattari M. Swarm intelligence // Scholarpedia. – 2007. – № 2 (9).
9. Yi Liu, Junyao Gao, Xuanyang Shi and Chunyu Jiang Decentralization of Virtual Linkage in Formation Control of Multi-Agents via Consensus Strategies Reprinted from: Applied Sciences 2018, 8, 2020
10. Eric Bonabeau, Marco Dorigo, and Guy Theraulaz, Swarm Intelligence from Natural to Artificial Systems, Oxford University Press, New York, 1999.
11. James Kennedy and Russell C. Eberhart, Swarm Intelligence, Academic Press, San Francisco, 2001.
12. S. Windnall, J. Peraire, Relative Motion using Rotating Axes, Lecture Notes in Dynamics, MIT, Boston, USA, 2008  
[http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-07-dynamics-fall-2009/lecture-notes/MIT16\\_07F09\\_Lec08.pdf](http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-07-dynamics-fall-2009/lecture-notes/MIT16_07F09_Lec08.pdf)
13. Yi Liu, Junyao Gao, Cunqiu Liu, Fangzhou Zhao and Jingchao Zhao Reconfigurable Formation Control of Multi-Agents Using Virtual Linkage Approach Reprinted from: Applied Sciences 2018, 8, 1109
14. Hölldobler B., Wilson E. O. The Ants. — Harvard University Press, 1990. — 732 p
15. Жураковський Ю.П. Теорія інформації та кодування: Підручник / Ю.П. Жураковський, В.П. Полторак. - К.: Вища шк, 2001. - 255 с. іл. Стор. 13.
16. S. Bouabdallah, Design and control of quadrotors with application to autonomous flying, EPFL, Lausanne, Switzerland, 2007
17. Е.Л. Ніколаї, Теорія гіроскопів, ОГІЗ, Москва, Росія, 1948
18. S. Windnall, J. Peraire, Relative Motion using Rotating Axes, Lecture Notes in Dynamics, MIT, Boston, USA, 2008  
[http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-07-dynamics-fall-2009/lecture-notes/MIT16\\_07F09\\_Lec08.pdf](http://ocw.mit.edu/courses/aeronautics-and-astronautics/16-07-dynamics-fall-2009/lecture-notes/MIT16_07F09_Lec08.pdf)
19. D. Kleppner, R.J. Kolenkow, An Introduction to Mechanics, McGraw-Hill, New York, USA, 1973
20. S. Bouabdallah, Design and control of quadrotors with application to autonomous flying, EPFL, Lausanne, Switzerland, 2007
21. R. W. Beard, Quadrotor dynamics and control, Lecture notes, Brigham Young University, Provo, USA,

2008, <http://rwbclasses.groups.et.byu.net/lib/exe/fetch.php?media=quadrotor:beardsquadratornotes.pdf> access 01.07. 2013

22. Eric Bonabeau, Marco Dorigo, and Guy Theraulaz, Swarm Intelligence from Natural to Artificial Systems, Oxford University Press, New York, 1999.

23. Li W. Stability analysis of swarm with general topology // IEEE Trans. Syst. Man. Cybern. B. 2008. Vol. 38. No. 4. P. 1084–1097.

24. Полторак В.П. Криптографічний захист даних в цифрових інформаційних системах (Частина 1) / В.П. Полторак // "Телеком. Військовий зв'язок. hi-Tech.ua." Спеціальний випуск. - Київ: Издательский дом СофтПресс, 2018. №2.- С. 98-104.

25. Виксин И.И. Модель обеспечения информационной безопасности киберфизических систем // Наука и бизнес: пути развития. 2018. № 2(80). С. 15–20.

## ДОДАТОК А

### Код скрипта для синхронного переміщення рою

```
clear all;
false = 0;
true = 1;

make_movie = true;
make_pot     = false;
writerObj = [];

%% Визначаємо площину точок для оцінки потенційної функції
x = linspace(0, 50, 100);
y = linspace(0, 50, 100);
z = 5;

[X,Y] = meshgrid(x,y);
[rx, cx] = size(X);
Z = zeros(rx, cx);

if( make_movie )
    writerObj = VideoWriter('3D_test3.avi');
    writerObj.FrameRate = 15;
    open(writerObj);
```

Зм	Лист	№ докум.	Підп.	Дата

**IA-62.080БАК.002**

Аркуш

97

```

end

az = -60;
el = 20;

azf = -60;
elf = 20;

%% Характеристики параметрів БПЛА
Nd = 1+20; % кількість дронів у рої
ind_c = -1; % індекс центрального/лідуючого агента
radius = 2; % радіус для безпілотників та можливих
перешкод
dims = [0, 50; 0, 50; 0, 50];
xc = [40, 40, 40]'; % початкова позиція центрального/лідуючого агента
end_loc = [5, 5, 5]'; % Бажане кінцеве розташування

%% Налаштування об'єкта шляху проходження
dist_thresh = 2; % Поріг відстані, який рій повинен бути в межах точки
проходження
wypt = Waypoints( dist_thresh );
wypt.addPoint( [ 15; 50; 20 ] ); % перша точка
wypt.addPoint( [ 0; 50; 0 ] ); % друга точка
wypt.addPoint( end_loc ); % кінцева точка

wypt2 = Waypoints( dist_thresh );
wypt2.addPoint( [ 15; 50; 20 ] ); % перша точка
wypt2.addPoint( [ 0; 50; 0 ] ); % друга точка
wypt2.addPoint( end_loc ); % кінцева точка

%% Характеристики параметрів для перешкод
No = 20;
i = 1;
obst = [];
dims_o = [0, 10; 0, 10; 0, 10];
while( i <= No )
    % випадково генерувати положення перешкоди в межах
    pos = rand(3,1).*( dims_o(:,2)-dims_o(:,1) ) + dims_o(:,1);

    % Додати нову перешкоду до масиву перешкод obst
    obst = [obst, Obstacle(pos,radius)];
    i = i + 1;
end

%% Створити дрони та масиви перешкод
i = 1;
drones = [];
while( i <= Nd )
    if( i == ind_c )
        % Додати лідуючий дрон до масиву
        drones = [drones, Drone(radius, xc , 2)];
    else
        % Додати слідкуючий дрон
        DEL = [xc - 10, xc + 10]; % змусити рій ініціалізуватися навколо
лідуючого дрона
        pos = rand(3,1).*( DEL(:,2)-DEL(:,1) ) + DEL(:,1);
        drones = [drones, Drone(radius, pos, 1)];
    end
    i = i + 1;
end
end

```



```

close all;
h = figure('Position', [10, 10, 1000, 800]);
set(gca, 'nextplot', 'replacechildren');
set(gcf, 'Renderer', 'zbuffer');
dims2 = dims';

i = 1;

figure(1)
N = 1;
if( make_pot )
    N = 2;
end

subplot(1,N,1)
while( i <=Nd )
    drawObject(drones(i));
    if( i == 1 )
        hold on
    end
    i = i + 1;
end

i = 1;
while( i <= No )
    drawObject(obst(i));
    i = i + 1;
end
hold off
grid on
view(az, el)
axis(dims2(:))
xpos = [];

it = 1;
count = 0;
done = 0;
while( done == 0 )
    i = 1;

    pt = wypt.getWaypoint( drones(1).pos );
    while( i <= Nd )
        drones(i).pos = drones(i).pos + GradientDescentUpdate(drones(i).pos,
i, drones, obst, pt );

        if( count > 20 || it >= 150 )
            done = 1;
        end
        i = i + 1;
    end

    % Відобразити дронів у новому положені
    i = 1;

    figure(1)
    subplot(1,N,1)

    while( i <= Nd )

```

```

        drawObject(drones(i));
        if( i == 1 )
            hold on
        end
        i = i + 1;
    end

    drawWaypoints( wypt2 );

    i = 1;
    while( i <= No )
        drawObject(obst(i));
        i = i + 1;
    end
    hold off
    coef = it/100;

    grid on
    axis(dims2(:))
    view(az + coef*(azf - az) , el + coef*(elf - el) )

    pause(.01)

    if( make_movie && it > 3 )
        ind = 1;
        if( N > 1 )
            subplot(1, N, 2)
            ind = 1;
            for ii = 1:rx
                for jj = 1:cx
                    Z(ii, jj) = getTotalPotential( [X(ii,jj); Y(ii, jj);
z(1)], ind, drones, obst, end_loc );
                end
            end

            surf(X,Y,Z)
            axis( [x(1), x(end), y(1), y(end)] )
        end

        frame = getframe(gcf);
        writeVideo(writerObj,frame);
    end

    it = it + 1;
end

if( make_movie )
    close(writerObj);
end

```

ДОДАТОК Б  
Код Скрипту для кластеризації

```
numRobots = 20;  
  
env = MultiRobotEnv(numRobots);  
  
env.robotRadius = 0.15;  
  
env.showTrajectory = false;  
  
  
detectors = cell(1,numRobots);
```

```

for rIdx = 1:numRobots

    detector = RobotDetector(env,rIdx);

    detector.maxDetections = numRobots;

    detector.maxRange = 10;

    detector.fieldOfView = pi/2;

    detectors{rIdx} = detector;

end

env.plotSensorLines = false;

numTeams = 5;

env.robotColors = repmat(hsv(numTeams),[ceil(numRobots/numTeams) 1]);

env.robotColors = env.robotColors(1:numRobots,:);


sampleTime = 0.1;

tVec = 0:sampleTime:25;


poses = [10*(rand(2,numRobots) - 0.5); ...
         pi*rand(1,numRobots)];

angleBias = 2*pi*(1:numRobots)/numTeams;

poses(1:2,:) = poses(1:2,:) + 2.5*[sin(angleBias);cos(angleBias)];


vel = zeros(3,numRobots);

for idx = 2:numel(tVec)

    env(1:numRobots, poses);

    xlim([-8 8]);

```

```

ylim([-8 8]);

for rIdx = 1:numRobots
    detections = step(detectors{rIdx});
    vel(:,rIdx) = swarmTeamController(poses,rIdx,detections,numTeams);
end

    poses = poses + vel*sampleTime;

end

function vel = swarmTeamController(poses,rIdx,detections,numTeams)

    pose = poses(:,rIdx);
    teamIdx = mod(rIdx,numTeams);

    % If there are no detections, turn in place
    v = 0;
    w = 2;

    if ~isempty(detections)
        validInds = find(mod(detections(:,3),numTeams) == teamIdx);
        if ~isempty(validInds)

```

```
range = mean(detections(validInds,1));
```

```
angle = mean(detections(validInds,2));
```

```
if range > 0.6
```

```
    v = 0.5;
```

```
elseif range < 0.4
```

```
    v = -0.5;
```

```
end
```

```
if angle > pi/12
```

```
    w = 2;
```

```
elseif angle < -pi/12
```

```
    w = -2;
```

```
end
```

```
end
```

```
end
```

```
vel = bodyToWorld([v;0;w],pose);
```

```
end
```

